ROYAL AIRCRAFT ESTABLISHMENT

U.D.C. 534.88 : 621.396.677.3 : 623.983 : 7.05 : 681.31

ROYAL AIRCRAFT ESTABLISHMENT

Technical Report 70130

July 1970

COMPUTER-AIDED DESIGN OF SONAR ARRAYS FOR MINIMUM SIDE-LOBE LEVEL

by

P. W. James

SUMMARY

The Report describes a way of designing a two dimensional array to minimise the side-lobes while maintaining any given beamwidth of the radiation pattern in the array plane. The amplitudes and phases associated with the elements are adjusted by a modified gradient method which uses a linear programming procedure. An example is given in which the side-lobe level for a six-element array is lowered by 21 dB.

Departmental Reference: Rad 1036

CONTENTS

1 INTRODUCTION

1.1 We are concerned with arrays consisting of a fairly small number (N) of elements placed in a horizontal plane. Each element is omnidirectional and unaffected by its neighbours. The array polar diagram will be considered in the horizontal plane only, and at a single frequency.

Having fixed, arbitrarily, the positions of the elements, then for a linear method of beam-forming, we have to choose 2N quantities, namely, the amplitude weighting and phase shift to be applied to each element. We normally desire that the final polar diagram should have a narrow main beam and low side-lobes. The present work deals with the problem which may be stated formally as follows:

For a given beamwidth, choose the weights and phases to minimise the largest side-lobe.

1.2 The Dolph-Tschebyscheff theory solves this problem for linear and equally spaced arrays. However, as far as I know, there is no theory available for the present case. The brute force method of trying all combinations of weights and phases is quite impossible, unless N is trivially small, since it requires far too much computer time. The only practicable method seems to be to select an arbitrary initial situation and then try to reduce the side-lobe level by some kind of gradient method.

In section 2 the method used in this Report is explained. It will be seen to be a gradient method, but with special treatment to deal with discontinuities in the gradient of the object function. The special treatment uses a linear programming procedure. In section 3 the method is applied to the array under consideration. In section 4 numerical results for a particular array are presented.

2 GENERAL METHOD

2.1 In this section we present in general terms the method of solution.

Any set of weights and phases may be represented as a vector in a space of 2N dimensions. Since the array pattern should have a preset beamwidth, the vectors are restricted to a space of dimension n, say, which is less than 2N. It might be expected that we could choose n coordinates arbitrarily, and solve for the remaining 2N - n. This turns out to be the case, so we can consider the problem to be one of minimising the side-lobe level when the point of interest is allowed to range freely over a space of n dimensions.

2.2 We shall denote the point of interest by $\underline{z} = (z_1, \ldots, z_n)$, the number of side-lobes by m and the values of the power at these side-lobes by $y_1, \ldots, y_m$. All these are functions of $\underline{z}$. It may be that for some $\underline{z}$, $m = 0$. There are then no side-lobes and the problem is solved. In general however $m > 0$ and so there exists a largest y_i with value $\hat{y}$:

$$\hat{y} = \max_i (y_i) \quad . \tag{1}$$

In order to reduce $\hat{y}$ it is natural to make a step in the direction of $-\text{grad } \hat{y}$

$$\Delta \underline{z} = - \varepsilon \text{ grad } \hat{y} \tag{2}$$

where $\varepsilon > 0$. The simplest rule is to set ε equal to a constant, and in fact the program to be described uses just this method in its initial stages. The flow chart is shown in Fig.1.

2.3 After a number of cycles of this simple gradient method, the procedure gets into difficulties. This happens when two or more side-lobes are nearly equal, and the identity of the largest changes from cycle to cycle. The gradient vector of $\hat{y}$ is discontinuous. It can happen that $\hat{y}$ is actually increased by a gradient step, due to the rôle of maximum side-lobe passing from one lobe to another. In the program it was arranged that if $\hat{y}$ failed to decrease over three successive iterations, the simple gradient method would be abandoned in favour of the linear programming method described below.

2.4 In the neighbourhood of the point of interest $\underline{z}_o$, we may suppose that the following linear approximations hold

$$y_i(\underline{z}) = y_i(\underline{z}_o) + (\underline{z} - \underline{z}_o) \cdot (\text{grad } y_i)_{\underline{z}_o} \quad . \tag{3}$$

Thus the change in y_i is proportional to the projection of the displacement $(\underline{z} - \underline{z}_o)$ on the gradient grad y_i. In order to change all the y_is, it is most efficient to use a displacement which belongs to the m-dimensional linear manifold spanned by the gradients, that is, for some scalars $c_1, \ldots c_m$,

$$(\underline{z} - \underline{z}_o)_j = \sum_{j=1}^{m} c_i (\text{grad } y_i)_j \tag{4}$$

or

$$\underline{z} - \underline{z}_o = \sum_{j=1}^{m} c_i \text{ grad } y_i \quad . \tag{5}$$

For, given any displacement, we could obtain the same changes in the y_is by using a smaller displacement, namely, that obtained by projecting the given displacement on to the manifold spanned by the gradients.

The point is that instead of dealing with an n-dimensional displacement, we need only consider the m-dimensional vector $\underline{c}$. Normally m is less than n, and so the amount of computation will be reduced. In matrix notation we put

$$\underline{z} - \underline{z}_o = \underline{g}' \, \underline{c} \tag{6}$$

where $\underline{g}$ is an $m \times n$ matrix whose ith row is grad y_i written as a row vector, and $\underline{c}$ is a column vector of order m. The original linearised equation may be written

$$\underline{y} = \underline{b} + \underline{g} \, (\underline{z} - \underline{z}_o) \tag{7}$$

where $\underline{y}$ and $\underline{b}$ are column n-vectors consisting of $y_i(\underline{z})$ and $y_i(\underline{z}_o)$. Substituting, we obtain

$$\underline{y} = \underline{b} + \underline{h} \, \underline{c} \tag{8}$$

where $\underline{h}$ is the $m \times m$ matrix $\underline{g} \, \underline{g}'$.

In order to avoid trouble with non-linearity, let us restrain $\underline{c}$ so that none of its components can numerically exceed a given quantity ϵ; that is,

$$|c_i| \leqslant \epsilon \quad , \qquad i = 1,2,\ldots,m \quad . \tag{9}$$

These can be converted into one-sided constraints by setting

$$c_i = p_i - q_i \; ; \qquad i = 1,\ldots,m \tag{10}$$

where the p_i, q_i satisfy

$$0 \leqslant p_i \leqslant \varepsilon \, , \quad 0 \leqslant q_i \leqslant \varepsilon \; ; \qquad i = 1,\ldots,m \, . \tag{11}$$

Further, we introduce r_i, s_i such that

$$p_i + r_i = \varepsilon \, , \quad q_i + s_i = \varepsilon \; ; \qquad i = 1,\ldots,m \tag{12}$$

where $r_i \geqslant 0$, $s_i \geqslant 0$; $\quad i = 1,\ldots,m$.
We now introduce yet more variables $x_i, \ldots, x_m$ such that

$$\hat{y} = y_i + x_i \; ; \qquad i = 1,\ldots,m \, . \tag{13}$$

Since $\hat{y} = \max_i (y_i)$, then $x_i \geqslant 0$ for each i. It is not convenient to have $\hat{y}$ appearing in more than one equation, so we shall retain the equation obtained from $i = 1$, namely

$$\hat{y} = y_1 + x_1 \tag{14}$$

and eliminate $\hat{y}$ from the rest

$$0 = y_i - y_1 + x_i - x_1 \; ; \quad i = 2,\ldots,m \, . \tag{15}$$

Substituting for y_i the value

$$b_i + \sum_{j=1}^{m} h_{ij} c_j \tag{16}$$

as given by the matrix equation (8), we obtain

$$\hat{y} = b_1 + \sum_{j=1}^{m} h_{1j}\, c_j + x_1 \tag{17}$$

$$0 = b_i - b_1 + x_i - x_1 + \sum_{j=1}^{m} (h_{ij} - h_{1j})\, c_j \; ; \qquad i = 2,\ldots,m \tag{18}$$

or

$$\hat{y} = b_1 + \sum_{j=1}^{m} h_{1j}\, (p_i - q_i) + x_1 \tag{19}$$

$$0 = b_i - b_1 + \sum_{j=1}^{m} (h_{1j} - h_{ij})\, (p_i - q_i) + x_i - x_1 \; ; \qquad i = 2,\ldots,m \tag{20}$$

2.5 To summarise, we now have a standard problem in linear programming, involving the (2m + 1) basic variables

$$x_1 \; ; \; p_1, \; \cdots \; , \; p_m \; ; \; q_1, \; \cdots \; , \; q_m$$

and the (3m − 1) slack variables

$$x_2, \; \cdots \; , \; x_m \; ; \; r_1, \; \cdots \; , \; r_m \; ; \; s_1, \; \cdots \; , \; s_m \; .$$

These are linked by the (3m − 1) equations

$$b_1 - b_i = x_i - x_1 + \sum_{j=1}^{m} (h_{ij} - h_{1j})\, p_j + \sum_{j=1}^{m} (- h_{ij} + h_{1j})\, q_j \; ; \; i = 2,\ldots,m \; , \qquad \ldots. \tag{21}$$

$$\varepsilon = r_i + p_i \; ; \qquad i = 1,\ldots,m \; , \tag{22}$$

$$\varepsilon = s_i + q_i \; ; \qquad i = 1,\ldots,m \; . \tag{23}$$

We have to maximise the object function $-\hat{y}$, given by

$$- b_1 = (-\hat{y}) + x_1 + \sum_{j=1}^{m} h_{1j} p_j + \sum_{j=1}^{m} - h_{1j} q_j \tag{24}$$

subject to the constraints

$$x_i \geq 0 , \quad p_i \geq 0 , \quad q_i \geq 0 , \quad r_i \geq 0 , \quad s_i \geq 0 ; \qquad i = 1,\ldots,m \ . \tag{25}$$

In solving this problem we may use the fact that a feasible solution is known, namely $p_i = q_i = r_i = s_i = 0$ corresponding to a zero step.

2.6 There are several ways of solving linear programming problems, but in this case we used the Simplex method. The tableau is shown as Table 1. The procedure will not be explained here, but readers not acquainted with it will find an elementary treatment in Vajda[1]. It is an algorithm which yields, after a finite number of steps, the values of all the variables and the maximised object function.

Having obtained p_i and q_i we calculate c_i from

$$c_i = p_i - q_i \tag{26}$$

and then by matrix multiplication find the step vector

$$\underline{z} - \underline{z}_o = \underline{g}' \, \underline{c} \ . \tag{27}$$

The (linearised) minimum value $\hat{y}_{lin}$ of $\hat{y}$ is obtained by negating the maximised object function of the linear programming routine.

Fig.2 is a simplified flowchart for the program.

2.7 We now calculate the actual value $\hat{y}_{true}$ of $\hat{y}$ at the new point of interest. If it were not for non-linearity this would be equal to the value $\hat{y}_{lin}$ given by the linear programming routine. If ε is small enough the two numbers will be nearly equal.

By repeated iteration we obtain a sequence of pairs of numbers

$$\hat{y}_{start} \ , \ (\hat{y}^{(1)}_{lin} \ , \ \hat{y}^{(1)}_{true}) \ , \ (\hat{y}^{(2)}_{lin} \ , \ \hat{y}^{(2)}_{true}) \ , \ \ldots$$

with the property

$$\hat{y}_{lin}^{(i+1)} \leqslant \hat{y}_{true}^{(i)} \qquad \text{for all } i. \tag{28}$$

If ϵ is small enough

$$\hat{y}_{lin}^{(i)} \simeq \hat{y}_{true}^{(i)} \tag{29}$$

so the sequence $\{\hat{y}_{true}^{(i)}\}$ is monotone decreasing, approximately. In order to guarantee a truly monotone decreasing sequence, we adopt the following procedure:-

"If $\hat{y}_{true}^{(i+1)} > \hat{y}_{true}^{(i)}$, reject the point (i + 1) just obtained, replace ϵ by $\epsilon/2$ and repeat the linear programming routine about the point of interest i".

It is intuitively obvious that halving ϵ, if necessary many times, will cause $\hat{y}_{true}^{(i+1)}$ to approach $\hat{y}_{lin}^{(i+1)}$, and since $\hat{y}_{lin}^{(i+1)}$ cannot exceed $\hat{y}_{true}^{(i)}$, we will eventually arrive at a value of $\hat{y}_{true}^{(i+1)}$ which is not greater than $\hat{y}_{true}^{(i)}$. Thus we obtain a sequence $\{\hat{y}_{true}^{(i)}\}$ which is genuinely monotone non-increasing, by this device of variable step length. The sequence is bounded below (by 0) and therefore it converges.

In practice it was found that the sequence converged quite rapidly (typically 20 iterations) until the values of $\hat{y}_{true}$ were constant apart from rounding errors. The values of ϵ did not approach zero. In the final steady state all the gradients grad y_i are zero, and all the y_is equal.

2.8 The variable-ϵ device was also used to speed up the convergence, by doubling ϵ when we appeared to be far from a final solution. The actual rule adopted was as follows:-

"Suppose $\hat{y}_{true}^{(i)}$, $\hat{y}_{true}^{(i+1)}$, $\hat{y}_{true}^{(i+2)}$ are three successive values of $\hat{y}$. Then if the difference between the last pair is more than half the difference between the first pair, i.e.

$$\hat{y}_{true}^{(i+2)} - \hat{y}_{true}^{(i+1)} > \tfrac{1}{2}\,(\hat{y}_{true}^{(i+1)} - \hat{y}_{true}^{(i)}) \tag{30}$$

we replace ϵ by 2ϵ on the next iteration". This rule allows us to start with a very small value of ϵ, say 10^{-6}. The computer will then keep

doubling ε until it approaches the final value, or until non-linearity effects cause the 'ε-halving' rule previously described to come into play.

No arrangements were made in the program to halt it when convergence was obtained. Successive values of $\hat{y}_{true}$ where printed and the program was interrupted by the operator when they became steady. The results were then obtained via common storage by entering an auxiliary program.

3 APPLICATION TO 2D ARRAY

3.1 Fig.3 shows the array consisting of N omnidirectional elements, which we label $1, 2, \ldots, N$, and in which element j has coordinates (x_j, y_j). The array may be regarded as lying in one plane (xy). We shall consider the polar diagram for directions lying only in this plane.

The 'array function' $f(A)$, which is a complex amplitude, is given by

$$f(A) = \sum_{j=1}^{N} w_j \exp ik (x_j \cos A + y_j \sin A) \tag{31}$$

where A = angle measured from the x-axis

$k = 2\pi$/wavelength

$w_1, \ldots, w_N$ are the (complex) weights associated with the elements.

The array power function will be defined to be

$$P(A) = |f(A)|^2 \ . \tag{32}$$

3.2 The problem may now be stated as follows:

Given N, k, $x_1, \ldots, x_N$, $y_N, \ldots, y_N$, choose $w_N, \ldots, w_N$ so that the side-lobe level is minimised, subject to the condition that the beamwidth has a prescribed value $2A_o$.

3.3 The restrictions on beamwidth will be taken to mean that the following equations hold

$$P(0) = 1$$

$$P(A_o) = \tfrac{1}{2} \ , \quad P(-A_o) = \tfrac{1}{2} \ . \tag{33}$$

Thus $2A_o$ is the -3 dB beamwidth (Fig.4).

The side-lobes will be defined as all the maxima of $P(A)$ that do not lie in the range $-A_o$ to A_o; i.e.

$$A = A_i \ , \qquad i = 1,\dots,m$$

where $\{A_i\}$ are the solutions of

$$\left.\begin{aligned} P'(A) &= 0 \\ P''(A) &< 0 \\ A_o < |A| &< \pi \end{aligned}\right\} . \tag{34}$$

The actual values of the side-lobes, the $\{y_i\}$ of the previous section, are

$$y_i = P(A_i) \tag{35}$$

3.4 The equation (33) may be expressed in terms of the array function as

$$\left.\begin{aligned} f(0) &= 1 \\ f(A_o) &= 2^{-\frac{1}{2}} \exp (i\ e_1) \\ f(-A_o) &= 2^{-\frac{1}{2}} \exp (i\ e_2) \end{aligned}\right\} \tag{36}$$

by introducing the (as yet unknown) phases e_1 and e_2. (No phase need be introduced in the first equation.) We then have three equations coupling the N variables $w_1,\dots,w_N$. Provided that N is at least 3, we can solve these equations (in general) for any 3 of $\{w_j\}$ in terms of e_1, e_2, $\{x_i\}$, $\{y_i\}$, A_o, k and the remaining $\{w_j\}$. We choose to solve for w_1, w_2 and w_3. There are then N-3 'free' variables $w_4,\dots,w_N$; or, rather, since $\{w_j\}$ are complex, there are 2N-6 free <u>real</u> variables at our disposal, to which must be added the e_1 and e_2, giving 2N-4 free variables. Thus we set $n = 2N-4$, and $z_1, z_2,\dots,z_n$ will correspond to

$$e_1, \quad e_2, \quad Re(w_4), \quad Im(w_4), \ldots, \quad Re(w_N), \quad Im(w_N) \quad . \tag{37}$$

The equations which express w_1, w_2, w_3 in terms of the free variables are cumbersome, but may be found in Appendix A.

3.5 Having forced the array function to satisfy the mainbeam conditions by solving the equations for w_1, w_2, w_3, we now need to locate the maxima. The method used was to compute $P(A)$ for every A at suitable intervals (say 10 deg) in order to find the maxima approximately, and then refine by solving

$$f'(A) = 0 \quad , \tag{38}$$

by Newton's method. The details are given in Appendix B.

All solutions with $|A| \leq A_o$ are then deleted and the remaining angles re-ordered so that $P(A_1)$ is the largest of the $P(A_i)$.

3.6 The procedures described in section 2 call for the gradients

$$\text{grad } P(A_i) \quad , \qquad\qquad i = 1,\ldots,m$$

taken with respect to the 2N-4 dimensional vector $\underline{z}$. Now

$$\text{grad } P(A_i) = \text{grad}_{A_i} P(A_i) + \frac{\partial P(A_i)}{\partial A_i} \text{grad } A_i \tag{39}$$

where grad_{A_i} denotes the gradient calculated as if A_i did not depend on $\underline{z}$. Fortunately, since A_i is a maximum,

$$\frac{\partial P(A_i)}{\partial A_i} = 0 \tag{40}$$

and so

$$\text{grad } P(A_i) = \text{grad}_{A_i} P(A_i) \quad . \tag{41}$$

The actual components will be

$$[\text{grad } P(A_i)]_j = \frac{\partial}{\partial e_j}[P(A_i)] ; \qquad j = 1,2 \tag{42}$$

$$[\text{grad } P(A_i)]_{2j+1} = \frac{\partial}{\partial(\text{Re}(w_{j+2}))}[P(A_i)] ; \qquad j = 1,\ldots,N-3 \tag{43}$$

$$[\text{grad } P(A_i)]_{2j+2} = \frac{\partial}{\partial(\text{Im}(w_{j+2}))}[P(A_i)] ; \qquad j = 1,\ldots,N-3 \quad . \tag{44}$$

When we carry out the differentiations, remembering that the free variables affect $P(A_i)$ not only explicitly but also via the variables w_1, w_2, w_3, the resulting expressions are rather lengthy, so these are relegated to Appendix C, (in which the components of $\text{grad } P(A_i)$ are written

$$d_1,\ d_2,\ g_1,\ g_2,\ldots,g_{2N-6}) \quad .$$

The numerical computation is not as long as the expressions might suggest since the computer can use much of previously stored information.

4 NUMERICAL EXAMPLE

4.1 The example uses six elements, placed regularly around a circle of radius 0.25 wavelength (Fig.5). The main beam is to be mid-way between two elements.

$$N = 6$$
$$x_1 = x_6 = 0.216506$$
$$x_2 = x_5 = 0$$
$$x_3 = x_4 = -0.216506$$
$$y_1 = y_3 = 0.125$$
$$y_2 = 0.25$$
$$y_4 = y_6 = -0.125$$
$$y_5 = -0.25$$
$$k = 2\pi$$

4.2 The most natural way to obtain a beam in the x-axis direction is to apply phases to bring the elements in-phase in this direction; and then use equal weighting amplitudes. This gives

$$w_j = \frac{1}{N} \exp(-\, i\, k\, x_j) \quad ; \quad j = 1,\ldots,N \qquad (45)$$

or

$$w_1 = w_6 = 0.034816 - 0.162990\, i$$
$$w_2 = w_5 = 0.166667$$
$$w_3 = w_4 = 0.034816 + 0.162990\, i \ .$$

The polar pattern of this array is plotted in Fig.6. The beamwidth is 84 deg. There are two side-lobes, located at ±162.3 deg with level -11.15 dB.

This pattern should be compared with the later results obtained by the side-lobe reduction program.

4.3 The high side-lobe levels under natural phasing makes this array a suitable subject for the program, provided we do not demand beamwidths much less than 84 deg. The actual values of $2A_o$ used were 55 deg to 90 deg in steps of 5 deg.

The initial values of the free variables were, at first, chosen to be

$$e_1 = 0, \quad e_2 = 0$$
$$w_4 = 0.0348162 + 0.162990\, i$$
$$w_5 = 0.166667$$
$$w_6 = 0.0348162 - 0.162990\, i \ ,$$

the last three being taken from the natural weighting. It was later found that this happened to be a rather unfavourable starting point, and machine time could be saved by starting from the values of e_1 to w_6 that constituted the final values for another A_o case.

Fig.7 illustrates the convergence of the maximum side-lobe as a plot against iteration number. The step length is also shown in Fig.8. For this

example, $2A_o = 85$ deg. The final power value was 4.40172×10^{-4} or -33.56 dB in this case, showing an improvement of over 20 dB compared with the natural weighting scheme for approximately the same beamwidth.

4.4 The results for the eight values of A_o are given in Tables 2 to 9. For each element, the tables give

(i) the x and y coordinates in wavelengths

(ii) the weights w_j in real and imaginary parts

(iii) the weights in polar form, that is, amplitude and phase, with the phase given in radians and in degrees.

The tables also contain the results of an independent program which computed the -3 dB points and listed the side-lobes.

The actual polar diagrams corresponding to these results are plotted in Figs.9 to 16.

It may be noted that the -3 dB points of these curves occur at the required angles. Further, all the side-lobes are at the same level. This common level depends on the beamwidth; the larger the beamwidth we can allow, the lower the side-lobe level. The trade-off between beamwidth and side-lobe level is illustrated in Fig.17, for this particular array. The point for the original phasing is also plotted on this figure, and it is about 21 dB above the curve. As the beamwidth increases, the side-lobe level drops rapidly. As the beamwidth decreases, the level increases as if to approach 0 dB at about 40^o. No solutions have been found which give reasonable patterns for beamwidths less than 50^o, which suggests that 'supergain' weightings do not exist for this particular array.

4.5 The tolerances for these arrays are also of interest. Naturally, reducing the side-lobe level makes the pattern more sensitive to phasing and other errors, as it is necessary to compare tolerances in a way which is not masked by this effect. In this Report we express the tolerance as the rms phase error which applied (independently) to all elements, leads to a variance of the complex array function equal to 0.001. (Roughly speaking, this means that the 'noise' in the pattern is 30 dB down.) The value can readily be shown to be

$$|f(0)|\left\{\frac{0.001}{\sum_{j=1}^{N}|w_j|^2}\right\}^{\frac{1}{2}}\cdot\frac{180}{\pi}\text{ degrees}\quad . \tag{46}$$

This tolerance is plotted in Fig.18 against beamwidth. On this curve is also shown the tolerance for natural phasing. Note that the tolerance is maximum at around 84°, the original beamwidth, and becomes less at narrower beamwidth, being about half the original value at 55°. For beamwidths around 70° to 90° the tolerance is only slightly less than the original tolerance with natural phasing (over 80%). The fact that the side-lobe levels are shown going down to some -40 dB does not of course mean that they could necessarily be achieved in practice. It would be necessary to consider, for example, mutual coupling, inter-element screening, departures from omnidirectionality, as well as the phase and amplitude tolerances. However these effects apply to any array, and it is outside the scope of this Report to include them.

4.6 The weights and phases for these optimised arrays have, in this example, rather curious properties. In the first place, they are not symmetric about the beam axis, that is

$$w_1 \neq w_6, \quad w_2 \neq w_5, \quad w_3 \neq w_4 \quad .$$

It follows that each solution is one of a pair, the other being obtained by reflection in the x-axis. Secondly, the array has conjugate complex symmetry about the y-axis, that is,

$$w_1 = w_3^*, \quad w_6 = w_4^*, \quad \text{and } w_2 \text{ and } w_5 \text{ are real.}$$

Probably these properties are due to the geometry of this example, but they show that these optimised arrays may be quite different from the arrays one would normally consider.

5 SUMMARY

5.1 We have described a method of adjusting the complex weights in a two-dimensional array in order to optimise its pattern. The optimisation consists of minimising the side-lobe level (that is, the largest side-lobe) while holding the -3 dB beamwidth at any value we choose.

The procedure is a gradient method, modified to deal with the discontinuities in the gradient vector. At each iteration we choose the next step using a linear programming routine. The step length is controlled by a parameter which is varied automatically so as to minimise the number of steps needed.

The method is a general one in that it can be applied to any problem requiring the minimising of the largest of several non-linear functions.

5.2 As an example, we considered a six-element array shaped like a regular hexagon, with the main beam mid-way between two elements. Starting from the natural phasing scheme, the program reduced the side-lobe level from -11 to -32 dB for unchanged beamwidth. The solutions indicated weights and phases strikingly different from the natural phasing scheme.

Figs.9 to 16 show the polar diagrams for this array for various beamwidths. The minimum side-lobe level decreases quickly as the beamwidth is allowed to increase, and Fig.17 shows the relation for this particular example. Such a curve may be used to select a beamwidth when the side-lobe level is prescribed.

The tolerance, expressed as the rms phase error which would produce pattern noise at -30 dB, were only slightly less than those for the original phasing (typically 80%), so there is no objection to the optimised arrays with regard to tolerance.

BLANK PAGE

Appendix A

A.1 We write the equations in matrix form

$$[B\ C]\begin{bmatrix} W \\ Z \end{bmatrix} = E \tag{A-1}$$

where $[B\ C]$ is the $6 \times 2N$ matrix of coefficients

$$\begin{bmatrix}
\cos b_1^o & -\sin b_1^o & \cos b_2^o & -\sin b_2^o & \cdots \\
\sin b_1^o & \cos b_1^o & \sin b_2^o & \cos b_2^o & \cdots \\
\cos b_1^+ & -\sin b_1^+ & \cos b_2^+ & -\sin b_2^+ & \cdots \\
\sin b_1^+ & \cos b_1^+ & \sin b_2^+ & \cos b_2^+ & \cdots \\
\cos b_1^- & -\sin b_1^- & \cos b_2^- & -\sin b_2^- & \cdots \\
\sin b_1^- & \cos b_1^- & \sin b_1^- & \cos b_1^- & \cdots
\end{bmatrix} \tag{A-2}$$

in which

$$b_j^o = k\,x_j \tag{A-3}$$

$$b_j^+ = k(x_j \cos A_o + y_j \sin A_o) \tag{A-4}$$

$$b_j^- = k(x_j \cos A_o - y_i \sin A_o)\ . \tag{A-5}$$

$\begin{bmatrix} W \\ Z \end{bmatrix}$ is the $2N$ column vector whose transpose is

$$[\mathrm{Re}(w_1),\ \mathrm{Im}(w_1),\ \mathrm{Re}(w_2),\ \mathrm{Im}(w_2),\ \ldots]\ . \tag{A-6}$$

E is the six-vector whose transpose is

$$[1,\ 0,\ (1/\sqrt{2})\cos e_1,\ (1/\sqrt{2})\sin e_1,\ (1/\sqrt{2})\cos e_2,\ (1/\sqrt{2})\sin e_2]\ . \tag{A-7}$$

The partitioning is such that B is 6×6, C is $6 \times (2N-6)$, w is 6×1, Z is $(2N-6) \times 1$. This allows us to write

$$BW + CZ = E \tag{A-8}$$

whence

$$W = B^{-1}(E - CZ) \tag{A-9}$$

or

$$W = AE - DZ \tag{A-10}$$

where A is the 6×6 matrix B^{-1} and D is the $6 \times (2N-6)$ matrix AC.

A depends on A_o and (x_j, y_i), $j = 1, 2, 3$; D depends on A_o and (x_j, y_j), $j = 4$ to $2N-6$; thus A and D are independent of the weights w_j and need not be recomputed at each iteration. The vector Z depends on w_j, $j = 4$ to $2N-6$, via the equation

$$\left.\begin{array}{l} Z_1 = \mathrm{Re}(w_4), \quad Z_2 = \mathrm{Im}(w_4) \\ \vdots \\ Z_{2j-7} = \mathrm{Re}(w_j), \quad Z_{2j-6} = \mathrm{Im}(w_j) \\ \vdots \\ Z_{2N-7} = \mathrm{Re}(w_N), \quad Z_{2N-6} = \mathrm{Im}(w_N) \end{array}\right\} . \tag{A-11}$$

The weights w_1, w_2, w_3 are then obtained by

$$\left.\begin{array}{l} w_1 = W_1 + i\,W_2 \\ w_2 = W_3 + i\,W_4 \\ w_3 = W_5 + i\,W_6 \end{array}\right\} . \tag{A-12}$$

A.2 There exists the possibility that B will be singular, in which case the method will fail. This will happen if, for example, $A_o = 0$. However for reasonable data no trouble has been encountered.

Appendix B

B.1 The first stage in locating the maxima begins by computing the array $p_1, \ldots, p_{M+1}$

where

$$\left.\begin{aligned} p_i &= P(\pi(2_i/M - 1)) , \qquad i = 1,\ldots,M \\ p_{M+1} &= p_1 \end{aligned}\right\} \qquad \text{(B-1)}$$

for some convenient integer M. M must be chosen large enough so that the interval $2\pi/M$ is less than the interval between maxima, and is found by experiment. We then examine the list and find all j such that

$$p_{i-1} < p_i \geqslant p_{i+1} \qquad \text{(B-2)}$$

giving as a first approximation

$$A = \pi(2_i/M - 1) \quad . \qquad \text{(B-3)}$$

B.2 The exact value of A at the maximum is then found by applying Newton's method to solve

$$P'(A) = 0 \quad . \qquad \text{(B-4)}$$

If A is an approximate solution, the next approximation A' is

$$A' = A - P'(A)/P''(A) \quad . \qquad \text{(B-5)}$$

The iteration is stopped when $|A' - A| < 10^{-4}$ and A' taken as the solution.

B.3 The formulae for P(A), P'(A) and P"(A) will now be derived.

Writing F_o and F_1 for the real and imaginary parts of $f(A)$, i.e.

$$F_o = \sum_{j=1}^{N} b_3, \quad F_1 = \sum_{j=1}^{N} b_4 \tag{B-6}$$

where

$$b_3 = Re(w_j) \cos b_o - Im(w_j) \sin b_o \tag{B-7}$$

$$b_4 = Re(w_j) \sin b_o + Im(w_j) \cos b_o \tag{B-8}$$

$$b_o = k\, x_j \cos A + k\, y_j \sin A \tag{B-9}$$

then

$$P(A) = F_o^2 + F_1^2 \ . \tag{B-10}$$

[The b_is depend on j, but the notation ignores this for simplicity.]

The first derivative $P'(A)$ is

$$P'(A) = 2\left\{F_o \frac{dF_o}{dA} + F_1 \frac{dF_1}{dA}\right\}$$

or

$$2(F_o F_2 + F_1 F_3) \tag{B-11}$$

where

$$F_2 = \sum_j \frac{db_3}{dA}$$

$$= \sum_j - b_4 \frac{db_o}{dA}$$

$$= \sum_j - b_4 b_1 \tag{B-12}$$

and

$$F_3 = \sum_j \frac{db_4}{dA}$$

$$= \sum_j b_3 b_1 \qquad \text{(B-13)}$$

where

$$b_1 = -k x_j \sin A + k y_j \cos A \quad . \qquad \text{(B-14)}$$

The second derivative $P''(A)$ is

$$P''(A) = 2\left\{F_o \frac{dF_2}{dA} + F_2 \frac{dF_o}{dA} + F_1 \frac{dF_3}{dA} + F_3 \frac{dF_1}{dA}\right\}$$

or

$$2\left\{F_o F_4 + F_1 F_5 + F_2^2 + F_3^2\right\} \qquad \text{(B-15)}$$

where

$$F_4 = \sum_j \frac{d}{dA} (-b_4 b_1)$$

$$= \sum_j \left[- b_4 \frac{db_1}{dA} - b_1 \frac{db_4}{dA}\right]$$

$$= \sum_j [- b_4 (- b_o) - b_1 (b_3 b_1)]$$

$$= \sum_j [b_4 b_o - b_3 b_1^2] \qquad \text{(B-16)}$$

and similarly

$$F_5 = \sum_j (- b_3 b_o - b_4 b_1^2) \quad . \qquad \text{(B-17)}$$

Appendix C

C.1 If α is any variable

$$\frac{\partial}{\partial\alpha} P(A_1) = 2(F_o G_o + F_1 G_1) \tag{C-1}$$

where

$$F_o = \sum_{j=1}^{N} \left[\mathrm{Re}(w_j) \cos b_o^{(j)} - \mathrm{Im}(w_j) \sin b_o^{(j)} \right] \tag{C-2}$$

$$F_1 = \sum_{j=1}^{N} \left[\mathrm{Re}(w_j) \sin b_o^{(j)} + \mathrm{Im}(w_j) \cos b_o^{(j)} \right] \tag{C-3}$$

$$b_o^{(j)} = k\, x_j \cos A + k\, y_j \sin A \tag{C-4}$$

and

$$G_o = \frac{\partial F_o}{\partial \alpha}, \quad G_1 = \frac{\partial F_1}{\partial \alpha} . \tag{C-5}$$

C.2 To evaluate d_1, take α to be e_1. Then

$$G_o = \frac{\partial}{\partial e_1} \sum_{j=1}^{N} \left[\mathrm{Re}(w_j) \cos b_o^{(j)} - \mathrm{Im}(w_j) \cos b_o^{(j)} \right] . \tag{C-6}$$

Now w_1, w_2, w_3 are functions of e_1 but the remaining weights are not. Hence

$$G_o = \sum_{j=1}^{3} \left\{ \frac{\partial}{\partial e_1} [\mathrm{Re}(w_j)] \cos b_o^{(j)} - \frac{\partial}{\partial e_1} [\mathrm{Im}(w_j)] \sin b_o^{(j)} \right\} . \tag{C-7}$$

However, $\mathrm{Re}(w_j)$ is the (2j-1)th component of W, and is equal to

$$\sum_{k=1}^{6} A_{2j-1,k} E_k - (DZ)_{2j-1} \tag{C-8}$$

while $\mathrm{Im}(w_j)$ is

$$\sum_{k=1}^{6} A_{2j,k} E_k - (DZ)_{2j} . \tag{C-9}$$

Differentiating w.r.t. e_1 gives

$$\frac{\partial}{\partial e_1} \operatorname{Re}(w_j) = \sum_{k=1}^{6} A_{2j-1,k} \frac{\partial E_k}{\partial e_1} \tag{C-10}$$

$$\frac{\partial}{\partial e_1} \operatorname{Im}(w_j) = \sum_{k=1}^{6} A_{2j,k} \frac{\partial E_k}{\partial e_1} \quad . \tag{C-11}$$

But

$$\frac{\partial E_k}{\partial e_1} = 0 \text{ unless } k = 3 \text{ or } 4, \text{ and } \frac{\partial E_3}{\partial e_1} = -E_4, \ \frac{\partial E_4}{\partial e_1} = E_3 \ . \tag{C-12}$$

Hence we get

$$\frac{\partial}{\partial e_1} \operatorname{Re}(w_j) = V_{1,2j-1} \tag{C-13}$$

$$\frac{\partial}{\partial e_1} \operatorname{Im}(w_j) = V_{1,2j} \tag{C-14}$$

where

$$V_{1,k} = -A_{k,3} E_4 + A_{k,4} E_3, \quad k = 1 \text{ to } 6 \quad . \tag{C-15}$$

Similarly, we write

$$\frac{\partial}{\partial e_2} \operatorname{Re}(w_j) = V_{2,2j-1} \tag{C-16}$$

$$\frac{\partial}{\partial e_2} \operatorname{Re}(w_j) = V_{2,2j-1} \tag{C-17}$$

where

$$V_{2,k} = -A_{k,5} E_6 + A_{k,6} E_5, \quad k = 1 \text{ to } 6 \quad . \tag{C-18}$$

By substitution

$$G_o = \sum_{j=1}^{3} \left[V_{k,2j-1} \cos b_o^{(j)} - V_{k,2j} \sin b_o^{(j)} \right] \tag{C-19}$$

and, similarly,

$$G_o = \sum_{j=1}^{3} \left[V_{k,2j-1} \sin b_o^{(j)} - V_{k,2j} \cos b_o^{(j)} \right] \tag{C-20}$$

where $k = 1$ for differentiation w.r.t. e_1, and 2 for e_2.

C.3 To find g_k we take α to be z_k. In differentiating F_o and F_1 we have to remember that w_1, w_2, w_3 depend on z_k via the matrix D, and also that one of $w_4, \ldots, w_{2N-6}$ depends directly on z_k.

If k is odd, say $2i - 7$ where $4 \leq i \leq N$, then

$$G_o = \sum_{j=1}^{3} \left[- D_{2j-1,k} \cos b_o^{(j)} + D_{2j,k} \sin b_o^{(j)} \right] + \cos b_o^{(i)} \tag{C-21}$$

$$G_1 = \sum_{j=1}^{3} \left[- D_{2j-1,k} \sin b_o^{(j)} - D_{2j,k} \cos b_o^{(j)} \right] + \sin b_o^{(i)} \quad . \tag{C-22}$$

If k is even, equal to $2i - 6$, then

$$G_o = \sum_{j=1}^{3} \left[- D_{2j-1,k} \cos b_o^{(j)} + D_{2j,k} \sin b_o^{(j)} \right] - \sin b_o^{(i)} \tag{C-23}$$

$$G_1 = \sum_{j=1}^{3} \left[- D_{2j-1,k} \sin b_o^{(j)} - D_{2j,k} \cos b_o^{(j)} \right] + \cos b_o^{(j)} \quad . \tag{C-24}$$

In either case, the gradient component is

$$g_k = 2(F_o G_o + F_1 G_1) \quad . \tag{C-25}$$

Table 1

TABLEAU FOR LINEAR PROGRAMMING ROUTINE

		z_1	p_1	....	p_m	q_1	....	q_m
z_2	$b_1 - b_2$	-1	$h_{21} - h_{11}$	....	$h_{2m} - h_{1m}$	$-h_{21} + h_{11}$	....	$-h_{2m} + h_{1m}$
z_3	$b_1 - b_3$	-1	$h_{31} - h_{11}$	....	$h_{3m} - h_{1m}$	$-h_{31} + h_{11}$	....	$-h_{3m} + h_{1m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
z_m	$b_1 - b_m$	-1	$h_{m1} - h_{11}$	....	$h_{mm} - h_{1m}$	$-h_{m1} + h_{11}$	....	$-h_{mm} + h_{1m}$
r_1	ε	0	1	....	0	0	....	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
r_m	ε	0	0	....	1	0	....	0
S_1	ε	0	0	....	0	1	....	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
S_m	ε	0	0	....	0	0		1
$-y_{max}$	$-b_1$	1	h_{11}	....	h_{1m}	$-h_{11}$	....	$-h_{1m}$

Table 2

BEAMWIDTH 55 DEG

N= 6 MAX.SL= .14Ø971 (-8.5Ø869 DB)
E1= 9.Ø71Ø3E-Ø3 E2=-8.73477E-Ø3 RAD

	1	2	3	4
X	.2165Ø6	Ø	-.2165Ø6	-.2165Ø6
Y	.125	.25	.125	-.125
WT-R	-.28ØØ5	.57Ø239	-.28Ø266	3.537Ø9E-Ø2
WT-I	3.Ø8759E-Ø2	-2.37417E-Ø4	-3.Ø3791E-Ø2	.19Ø323
WT-AMP	.281747	.57Ø239	.2819Ø8	.193582
PH-RAD	3.Ø3178	-4.16347E-Ø4	-3.Ø3362	1.387Ø5
PH-DEG	173.7Ø8	-2.38549E-Ø2	-173.814	79.4719

	5	6
X	Ø	.2165Ø6
Y	-.25	-.125
WT-R	.219784	3.52782E-Ø2
WT-I	-3.87944E-Ø6	-.19Ø247
WT-AMP	.219784	.19349
PH-RAD	-1.76512E-Ø5	-1.38744
PH-DEG	-1.Ø1134E-Ø3	-79.4948

1/2 PWR -27.5 27.5 DEG
Ø DB @ Ø DEG
-8.51 DB @ 1Ø1.3 DEG
-8.51 DB @ 18Ø DEG
-8.51 DB @ -1Ø1.3 DEG
DIR 4.31995 (6.35 DB)
-3ØDB TOLS:
POS 6.45862E-Ø3 WL
WT .352439 DB
PH 2.3251 DEG

Table 3

BEAMWIDTH 6Ø DEG

N= 6 MAX.SL= 6.78598E-Ø2 (-11.6839 DB)
E1= 4.96462E-Ø3 E2=-5.Ø2428E-Ø3 RAD

	1	2	3	4
X	.2165Ø6	Ø	-.2165Ø6	-.2165Ø6
Y	.125	.25	.125	-.125
WT-R	-.2Ø2399	.474223	-.2Ø235	5.35152E-Ø2
WT-I	-2.2Ø666E-Ø2	5.18891E-Ø5	2.19617E-Ø2	.167Ø44
WT-AMP	.2Ø3598	.474223	.2Ø3539	.1754Ø7
PH-RAD	-3.Ø3299	1.Ø9419E-Ø4	3.Ø3348	1.26Ø76
PH DEG	-173.778	6.26925E-Ø3	173.8Ø6	72.2363

	5	6
X	Ø	.2165Ø6
Y	-.25	-.125
WT-R	.218211	5.354ØØE-Ø2
WT-I	-1.Ø83Ø9E-Ø5	-.167Ø23
WT-AMP	.218211	.175394
PH-RAD	-4.9635ØE-Ø5	-1.26Ø59
PH-DEG	-2.84388E-Ø3	-72.2265

1/2 PWR -3Ø 3Ø DEG
Ø DB @ Ø DEG
-11.68 DB @ -1Ø5.7 DEG
-11.68 DB @ 1Ø5.7 DEG
-11.68 DB @ 18Ø DEG
DIR 4.86244 (6.87 DB)
-3ØDB TOLS:
POS 7.79463E-Ø3 WL
WT .425343 DB
PH 2.8Ø6Ø7 DEG

Table 4

BEAMWIDTH 65 DEG

N= 6 MAX.SL= 3.11699E-Ø2 (-15.Ø626 DB)
E1= 2.9Ø223E-Ø3 E2=-2.88797E-Ø3 RAD

	1	2	3	4
X	.2165Ø6	Ø	-.2165Ø6	-.2165Ø6
Y	.125	.25	.125	-.125
WT-R	-.127Ø46	.391772	-.127Ø47	5.51484E-Ø2
WT-I	-6.5Ø121E-Ø2	-3.34624E-Ø6	6.5Ø226E-Ø2	.145494
WT-AMP	.142714	.391772	.14272	.155595
PH-RAD	-2.66861	-8.54131E-Ø6	2.66855	1.2Ø849
PH-DEG	-152.9	-4.89381E-Ø4	152.897	69.2412

	5	6
X	Ø	.2165Ø6
Y	-.25	-.125
WT-R	.226542	5.51492E-Ø2
WT-I	-2.78361E-Ø6	-.145485
WT-AMP	.226542	.155587
PH-RAD	-1.22874E-Ø5	-1.2Ø846
PH-DEG	-7.Ø4Ø16E-Ø4	-69.2397

```
1/2 PWR  -32.5        32.5      DEG
 Ø         DB @   Ø         DEG
-15.Ø6     DB @  18Ø        DEG
-15.Ø6     DB @ -111.1      DEG
-15.Ø6     DB @  111.1      DEG
DIR   4.98824    ( 6.98     DB)
-3ØDB TOLS:
POS   9.28274E-Ø3   WL
WT    .5Ø6547       DB
PH    3.34178       DEG
```

Table 5

BEAMWIDTH 7Ø DEG

N= 6 MAX.SL= 1.333Ø8E-Ø2 (-18.7514 DB)
E1=-4.47159E-Ø4 E2= 3.25132E-Ø4 RAD

	1	2	3	4
X	.2165Ø6	Ø	-.2165Ø6	-.2165Ø6
Y	.125	.25	.125	-.125
WT-R	-6.98724E-Ø2	.334Ø35	-6.98346E-Ø2	6.12264E-Ø2
WT-I	-9.16232E-Ø2	6.Ø3756E-Ø5	9.147Ø1E-Ø2	.134584
WT-AMP	.115226	.334Ø35	.115Ø81	.147856
PH-RAD	-2.22231	1.8Ø746E-Ø4	2.22286	1.14385
PH-DEG	-127.329	1.Ø356ØE-Ø2	127.361	65.5378

	5	6
X	Ø	.2165Ø6
Y	-.25	-.125
WT-R	.227295	6.12369E-Ø2
WT-I	-4.Ø9895E-Ø6	-.134572
WT-AMP	.227295	.14785
PH-RAD	-1.8Ø336E-Ø5	-1.14375
PH-DEG	-1.Ø3325E-Ø3	-65.5321

1/2 PWR -35 35 DEG
Ø DB @ Ø DEG
-18.75 DB @ 117.1 DEG
-18.75 DB @ -117.1 DEG
-18.75 DB @ 18Ø DEG
DIR 4.81854 (6.83 DB)
-3ØDB TOLS:
POS 1.Ø4158E-Ø2 WL
WT .568377 DB
PH 3.74968 DEG

Table 6

BEAMWIDTH 75 DEG

N= 6 MAX.SL= 5 13755E-Ø3 (-22.8924 DB)
E1= 4.13922E-Ø4 E2=-4.2Ø647E-Ø4 RAD

	1	2	3	4
X	.2165Ø6	Ø	-.2165Ø6	-.2165Ø6
Y	.125	.25	.125	-.125
WT-R	-1.75293E-Ø2	.2881Ø4	-1.75279E-Ø2	6.21181E-Ø2
WT-I	-.11Ø457	3.13678E-Ø6	.11Ø447	.126866
WT-AMP	.111839	.2881Ø4	.111829	.141257
PH-RAD	-1.72818	1.08877E-Ø5	1.72818	1.11547
PH-DEG	-99.Ø175	6.23817E-Ø4	99.Ø175	63.912

	5	6
X	Ø	.2165Ø6
Y	-.25	-.125
WT-R	.2291Ø4	6.21188E-Ø2
WT-I	-8.76854E-Ø7	-.126864
WT-AMP	.2291Ø4	.141256
PH-RAD	-3.82732E-Ø6	-1.11546
PH-DEG	-2.19289E-Ø4	-63.9114

1/2 PWR -37.5 37.5 DEG
Ø DB @ Ø DEG
-22.89 DB @ -123.7 DEG
-22.89 DB @ 123.7 DEG
-22.89 DB @ 18Ø DEG
DIR 4.56Ø76 (6.59 DB)
-3ØDB TOLS:
POS 1.12424E-Ø2 WL
WT .613481 DB
PH 4.Ø4724 DEG

Table 7

BEAMWIDTH 8Ø DEG

N= 6   MAX.SL= 1.69646E-Ø3   (-27.7Ø45   DB)
E1=-1.84984E-Ø3   E2= 1.85Ø59E-Ø3   RAD

	1	2	3	4
X	.2165Ø6	Ø	-.2165Ø6	-.2165Ø6
Y	.125	.25	.125	-.125
WT-R	2.73322E-Ø2	.253Ø21	2.73332E-Ø2	6.28844E-Ø2
WT-I	-.12Ø433	3.5296ØE-Ø6	.12Ø423	.124688
WT-AMP	.123496	.253Ø21	.123486	.139648
PH-RAD	-1.34763	1.39498E-Ø5	1.3476	1.1Ø369
PH-DEG	-77.2134	7.99266E-Ø4	77.2119	63.2367

	5	6
X	Ø	.2165Ø6
Y	-.25	-.125
WT-R	.22988	6.28856F-Ø2
WT-I	-3.63328E-Ø6	-.124677
WT-AMP	.22988	.139639
PH-RAD	-1.58Ø51E-Ø5	-1.1Ø364
PH-DEG	-9.Ø5567E-Ø4	-63.2342

```
1/2 PWR  -4Ø    4Ø    DEG
 Ø          DB @  Ø          DEG
-27.7       DB @  13Ø.8      DEG
-27.7       DB @ -13Ø.8      DEG
-27.7       DB @  18Ø        DEG
DIR  4.28581    ( 6.32       DB)
-3ØDB TOL.:
POS   1.16584E-Ø2   WL
WT    .636183       DB
PH    4.197Ø1       DEG
```

Table 8

BEAMWIDTH 85 DEG

N= 6 MAX.SL= 4.4Ø3Ø8E-Ø4 (-33.5624 DB)
E1=-3.17555E-Ø6 E2= 3.21454E-Ø6 RAD

	1	2	3	4
X	.2165Ø6	Ø	-.2165Ø6	-.2165Ø6
Y	.125	.25	.125	-.125
WT-R	6.71199E-Ø2	.22631	6.71194E-Ø2	6.28798E-Ø2
WT-I	-.1256Ø9	3.11Ø32E-Ø6	.125597	.12468
WT-AMP	.142417	.22631	.142407	.139639
PH-RAD	-1.Ø8ØØ4	1.37436E-Ø5	1.Ø8ØØ1	1.1Ø369
PH-DEG	-61.8819	7.87452E-Ø4	61.88	63.2369

	5	6
X	Ø	.2165Ø6
Y	-.25	-.125
WT-R	.229864	6.288Ø3E-Ø2
WT-I	-3.937Ø7E-Ø6	-.124669
WT-AMP	.229864	.139629
PH-RAD	-1.71278E-Ø5	-1.1Ø365
PH-DEG	-9.81352E-Ø4	-63.2346

1/2 PWR -42.5 42.5 DEG
Ø DB @ Ø DEG
-33.56 DB @ 138.5 DEG
-33 DB @ -138.5 DEG
-3. DB @ 18Ø DEG
DIF 36 (6.Ø5 DB)
-3ØDB TOLS:
POS 1.17455E-Ø2 WL
WT .64Ø936 DB
PH 4.22837 DEG

Table 9

BEAMWIDTH 9Ø DEG

N= 6 MAX.SL= 1.Ø7Ø51E-Ø4 (-39.7Ø41 DB)
E1= 4.18666E-Ø4 E2=-4.1768ØE-Ø4 RAD

	1	2	3	4
X	.2165Ø6	Ø	-.2165Ø6	-.2165Ø6
Y	.125	.25	.125	-.125
WT-R	.1Ø2Ø1	.2Ø48Ø6	.1Ø2Ø15	6.3436ØE-Ø2
WT-I	-.126915	6.Ø6649E-Ø6	.1269Ø7	.126Ø92
WT-AMP	.162829	.2Ø48Ø6	.162826	.14115
PH-RAD	-.893764	2.962Ø6E-Ø5	.89371	1.1Ø468
PH-DEG	-51.2Ø89	1.69714E-Ø3	51.2Ø59	63.2934

	5	6
X	Ø	.2165Ø6
Y	-.25	-.125
WT-R	.231238	6.34371E-Ø2
WT-I	-3.Ø7767E-Ø6	-.126Ø81
WT-AMP	.231238	.141141
PH-RAD	-1.33Ø95E-Ø5	-1.1Ø464
PH-DEG	-7.6258ØE-Ø4	-63.291

1/2 PWR -45 45
Ø DB @ Ø DEG
-39.7 DB @ -149.4 DEG
-39.71 DB @ 149.4 DEG
-39.71 DB @ 18Ø DEG
DIR 3.79Ø87 (5.79 DB)
-3ØDB TOLS:
POS 1.15987E-Ø2 WL
WT .632929 DB
PH 4.17555 DEG

SYMBOLS

A	azimuthal angle
A_0	semi-beamwidth to -3 dB points
$\underline{b}$	$\underline{y}$, at beginning of iteration
b_i	components of $\underline{b}$
$\underline{c}$	vector consisting of $c_1,\ldots,c_m$
c_i	scalars expressing $\underline{\Delta z}$ in terms of the gradients
e_1, e_2	phases at -3 dB points
$f(.)$	complex array function
$\underline{g}$	matrix whose rows are grad $\hat{y}_i$
g_{ij}	components of $\underline{g}$
$\underline{h}$	$\underline{g}\ \underline{g}'$
h_{ij}	components of $\underline{h}$
i	$\sqrt{-1}$
i, j	indices
k	2π/wavelength
m	number of side-lobes
N	number of elements
n	number of free variables
$P(.)$	array power function $\|f(.)\|^2$
p_i, q_i, r_i, s_i	auxiliary variables used to convert problem to standard linear programming form
w_i	complex weight for element i
x_i, y_i	Cartesian coordinates of element i
y_i	power level of side-lobe i
$\hat{y}$	largest of y_i
y_{start}	initial value of $\hat{y}$
$\hat{y}^{(i)}_{lin}$	linearised $\hat{y}$ after iteration i
$\hat{y}^{(i)}_{true}$	true $\hat{y}$ after iteration i
$\underline{z}$	vector consisting of $z_1,\ldots,w_n$
z_i	free variables, identified with e_1, e_2 and the real and imaginary parts of $w_4,\ldots,w_N$

SYMBOLS (Cont'd)

$\underline{\Delta z}$	increment in $\underline{z}$ resulting from iteration
ε	small constant limiting step length
grad	gradient operator $\left(\frac{\partial}{\partial z_1}, \ldots, \frac{\partial}{\partial z_n}\right)$
$'$	matrix transpose

REFERENCE

No.	Author	Title, etc.
1	S. Vajda	Linear programming and the theory of games. Methuen & Co. Ltd., London (1960)

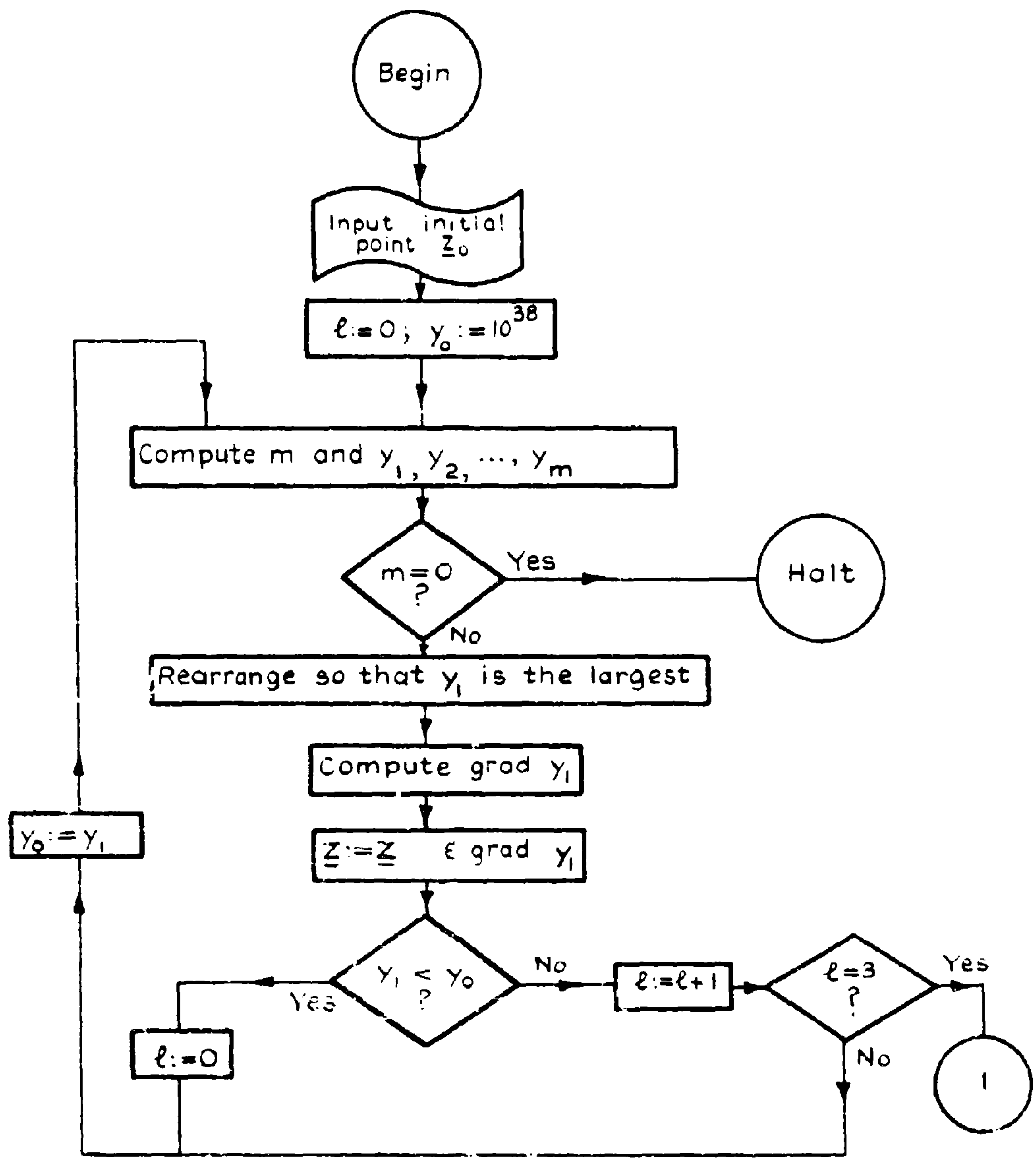

Fig. 1 Flow chart for first method of sidelobe reduction

Fig. 2

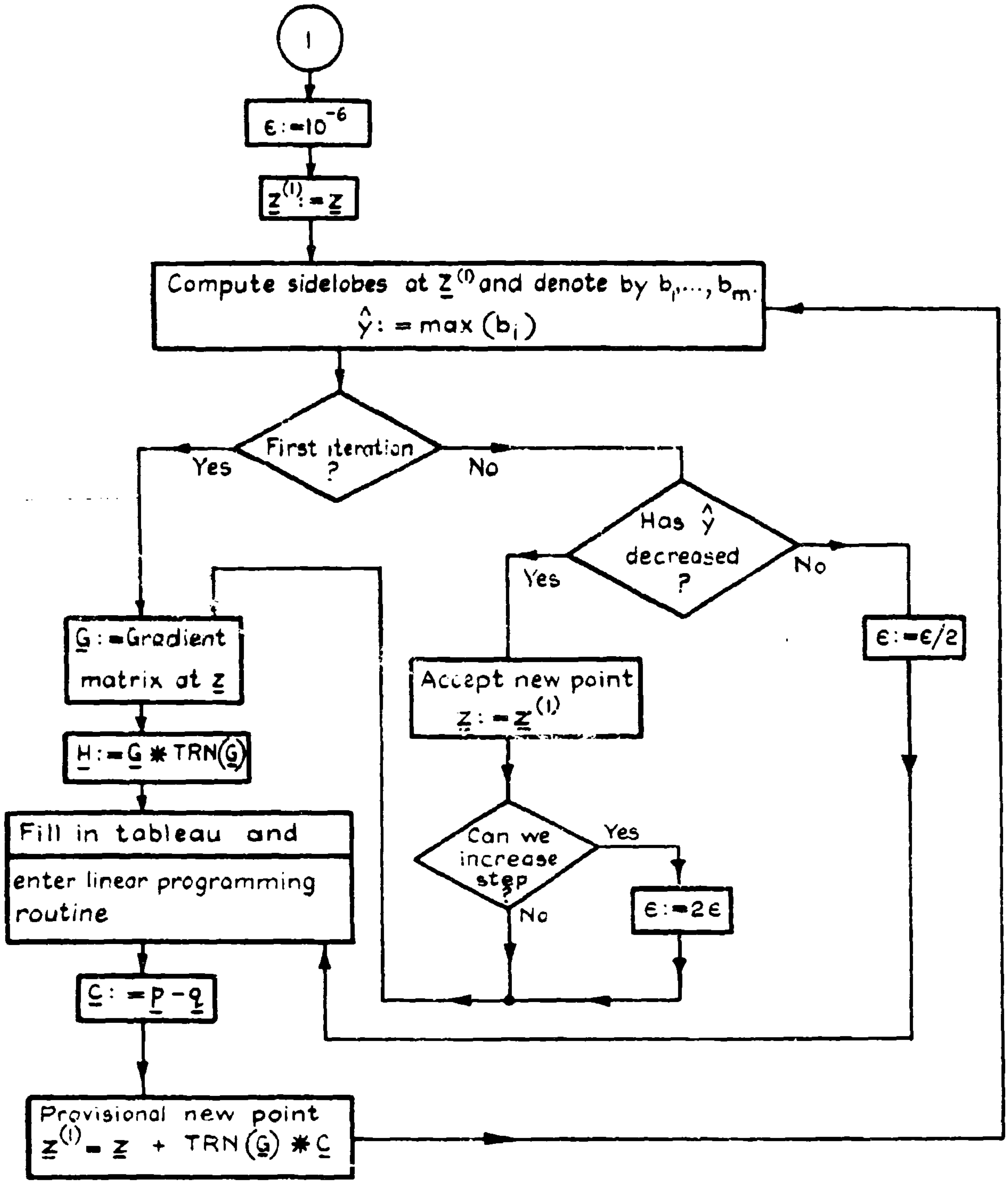

Fig. 2 Simplified flow chart for the second method

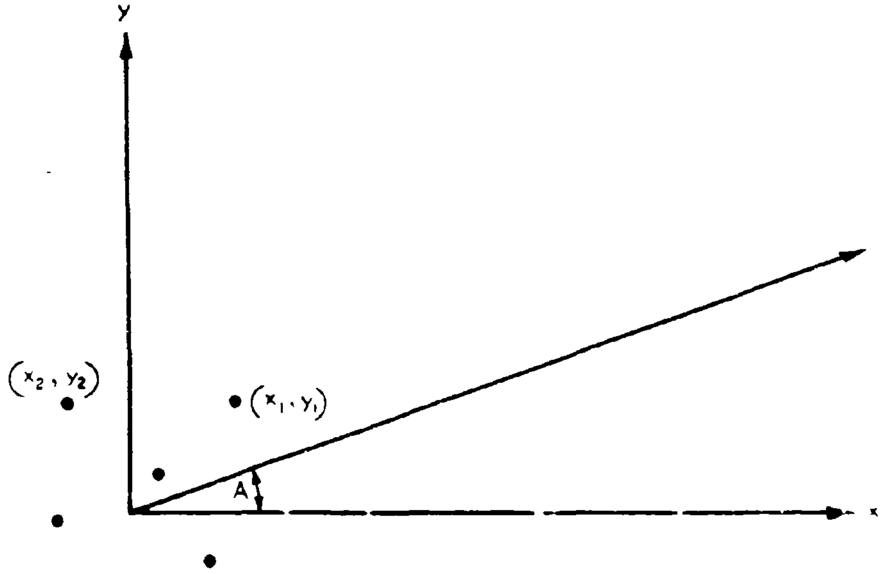

Fig. 3 Array configuration

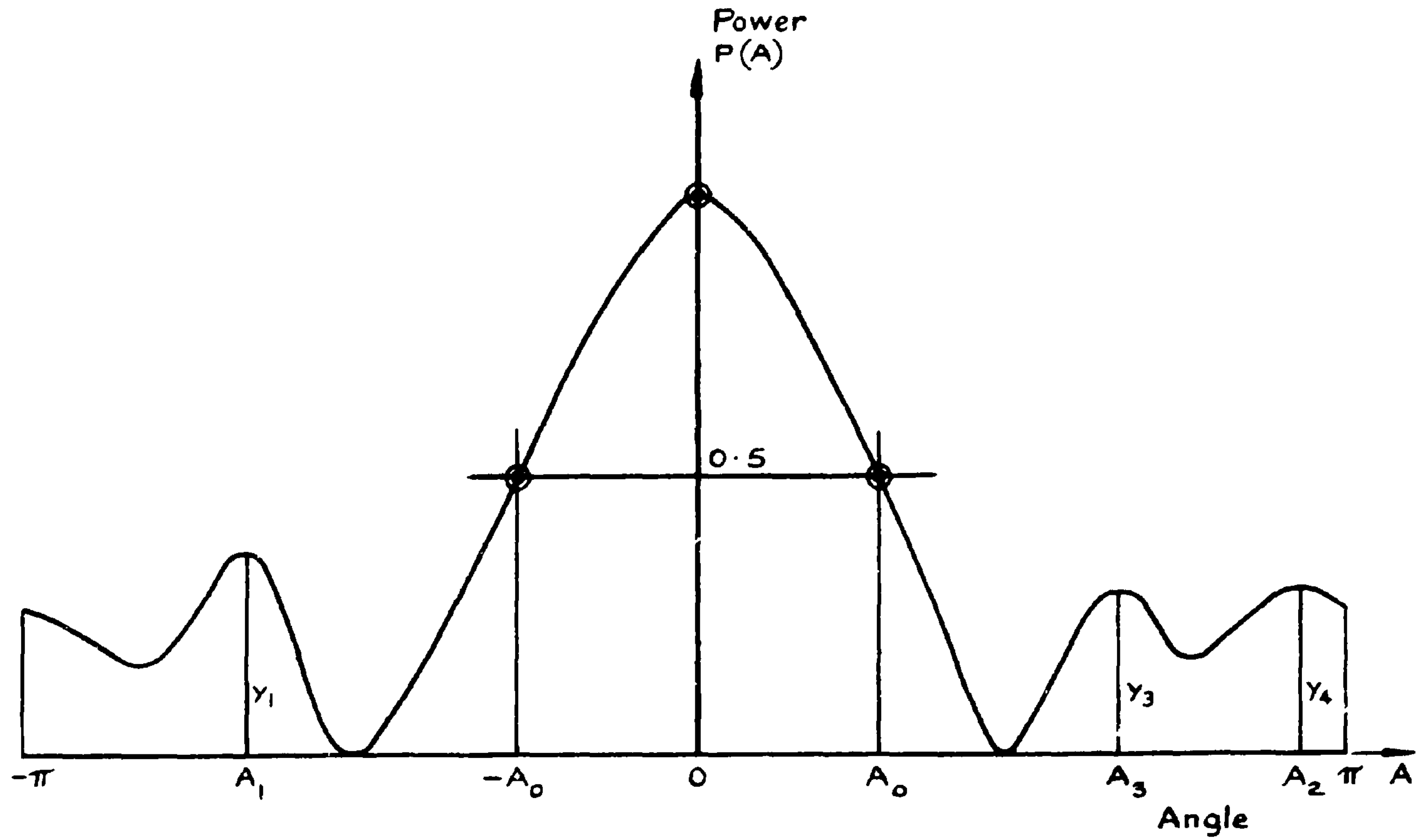

Fig. 4 Illustrating beam-defining points and sidelobe definitions

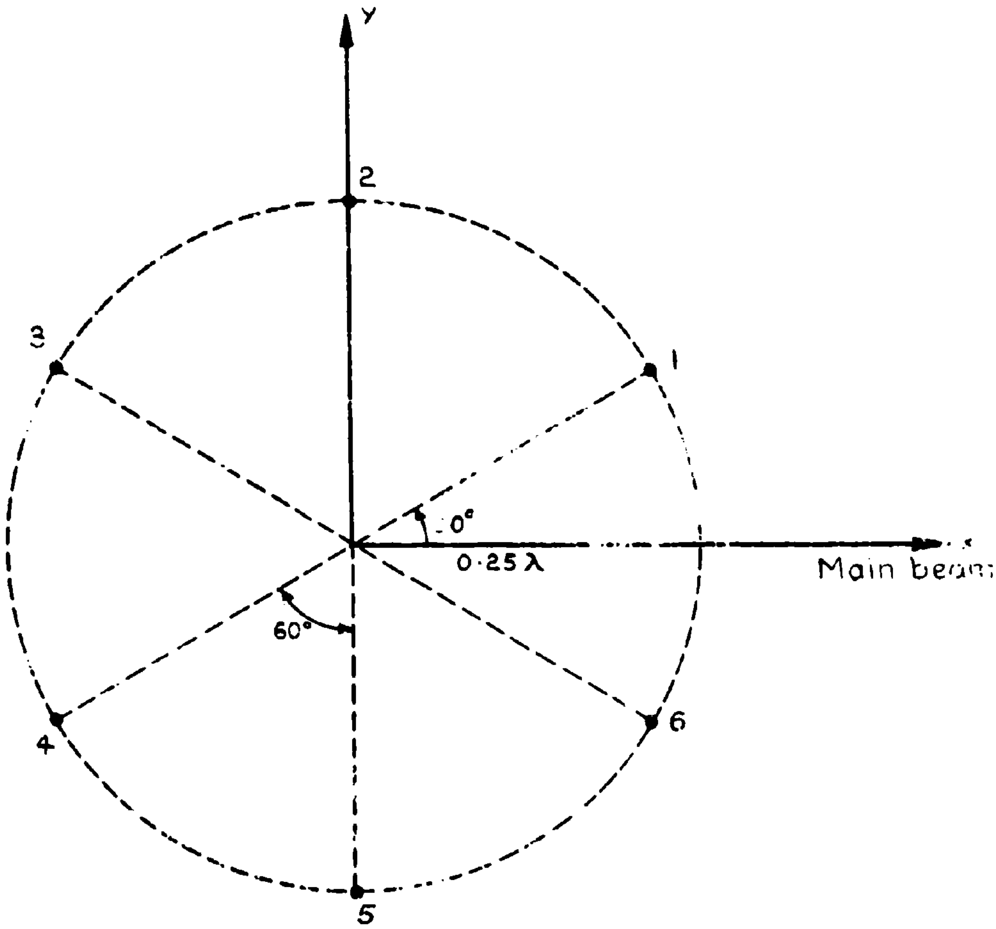

Fig. 5 Array used as example

Fig. 6

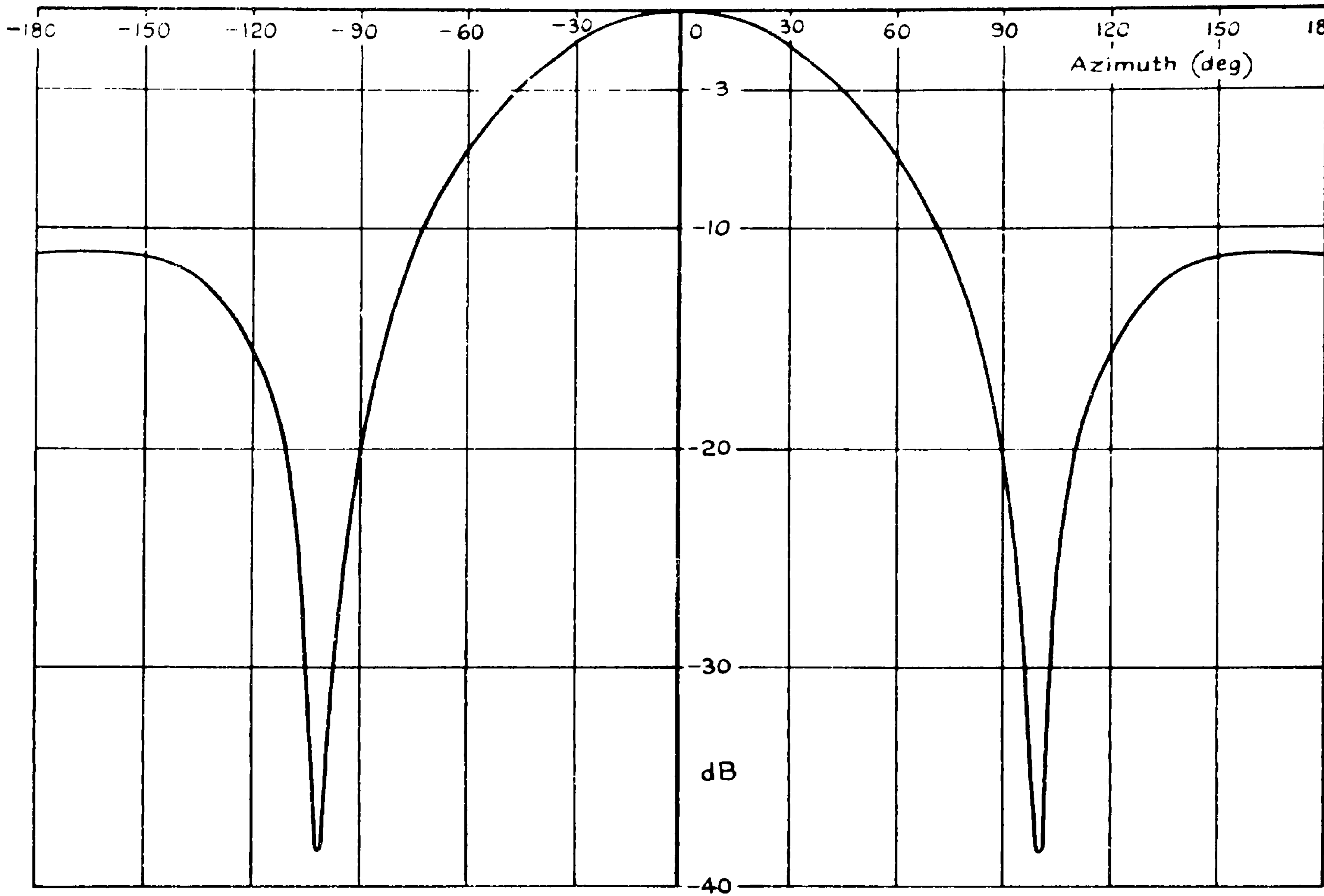

Fig. 6 Initial array pattern

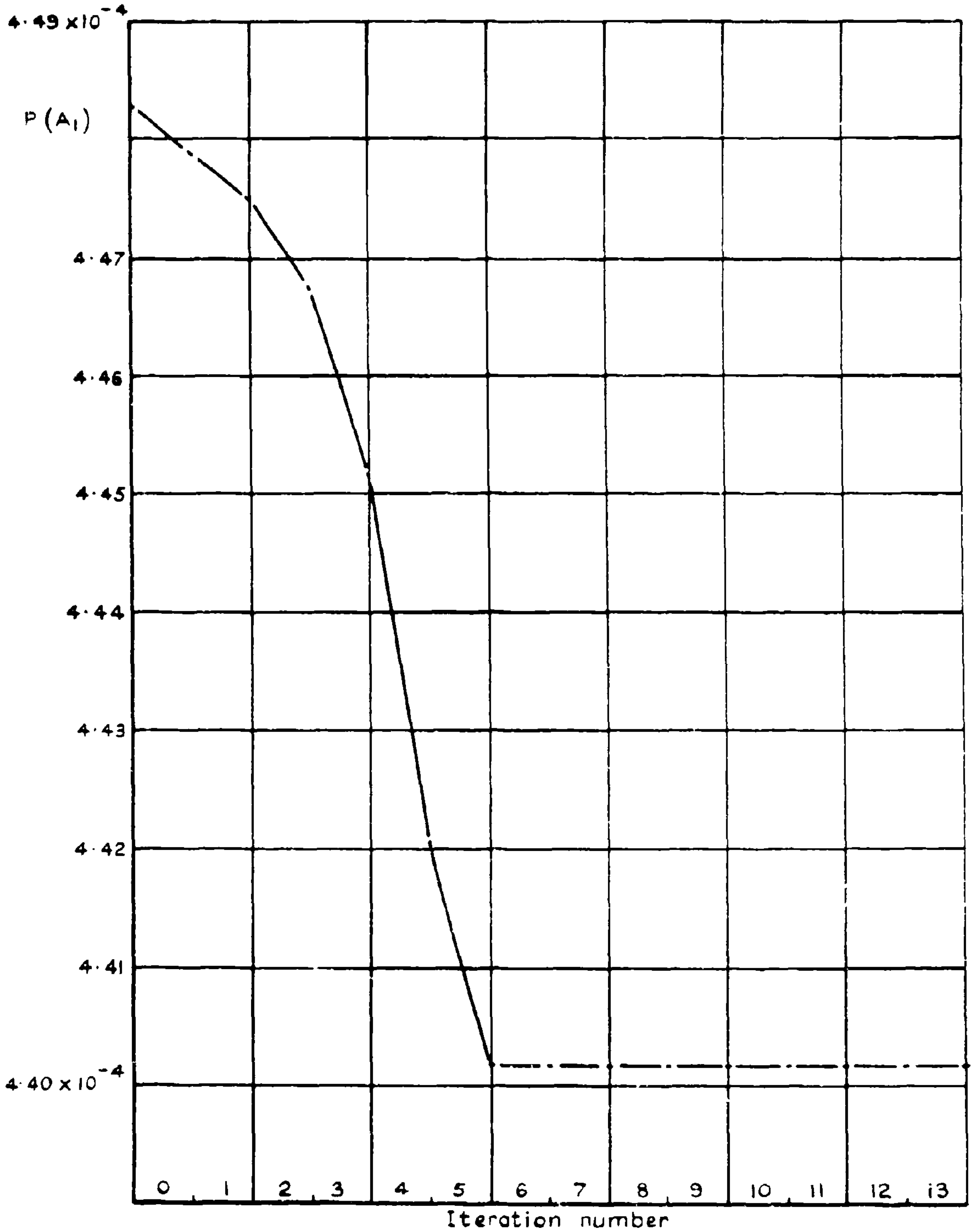

Fig.7 Typical sequence of $\hat{y}_{true}$ values during computation

Fig. 8

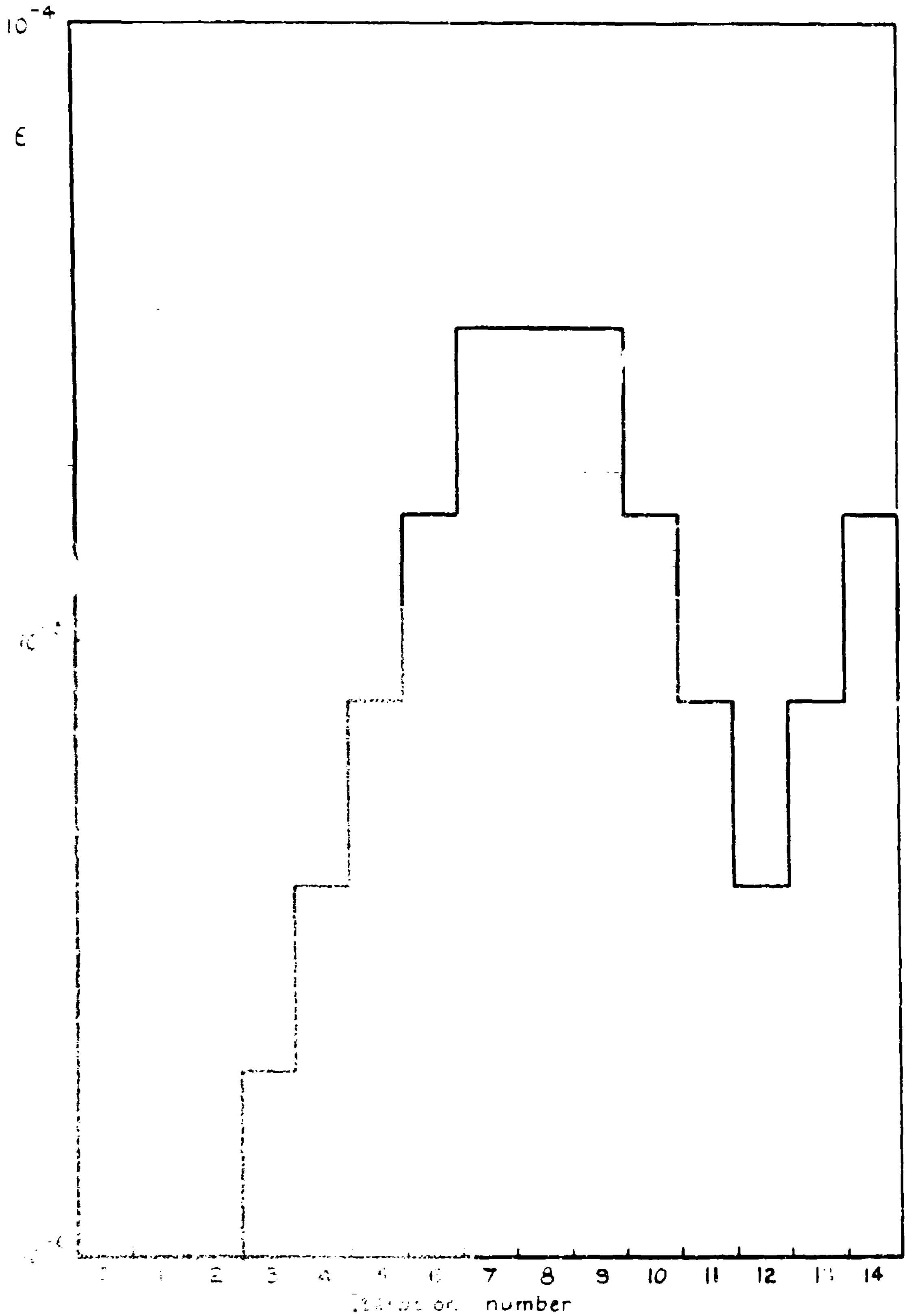

Fig 8 Typical sequence of ϵ values during computation

TR 70130

OOP 902483

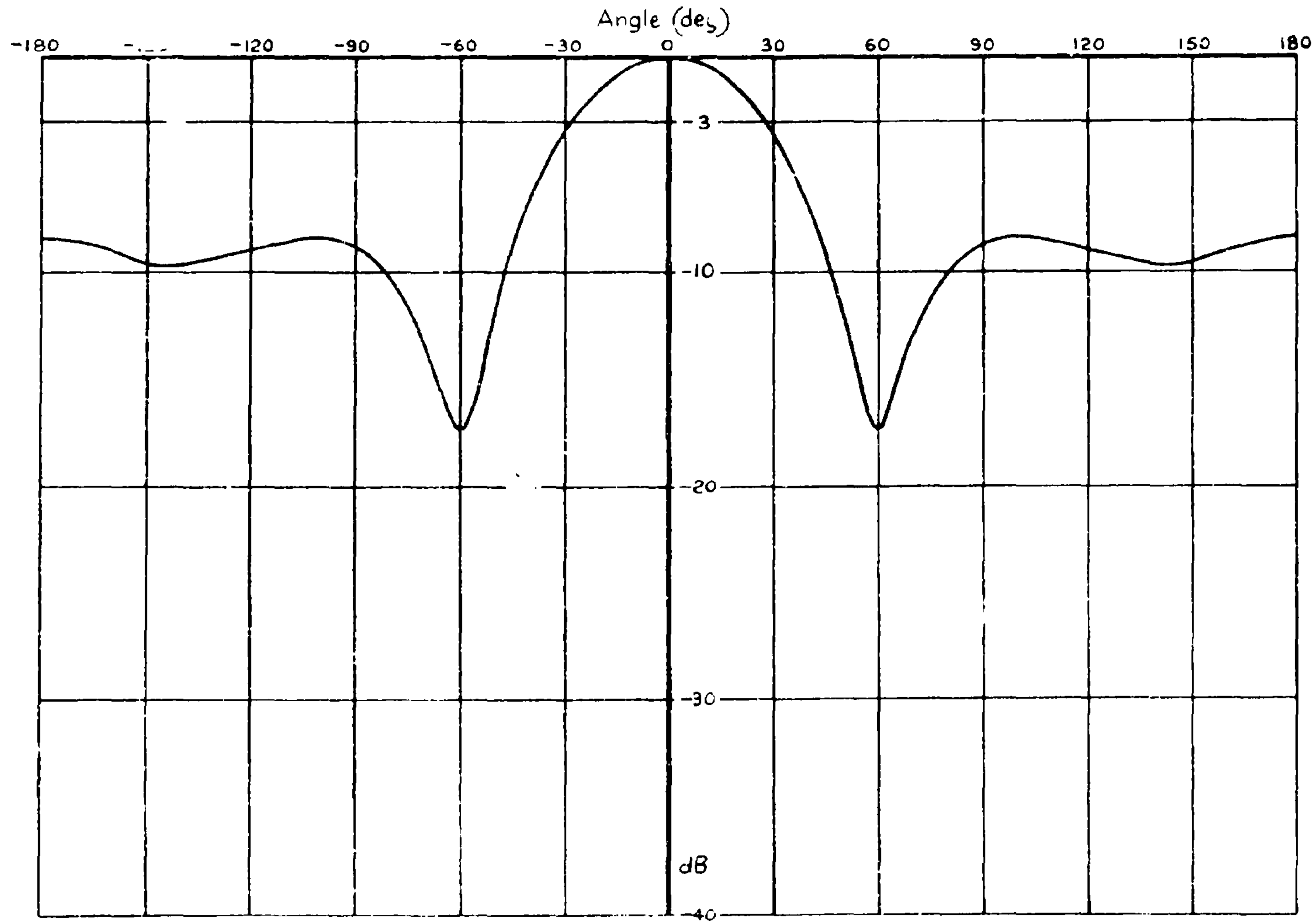

Fig. 9 Optimised pattern for 55° beamwidth

Fig. 9

Fig. 10

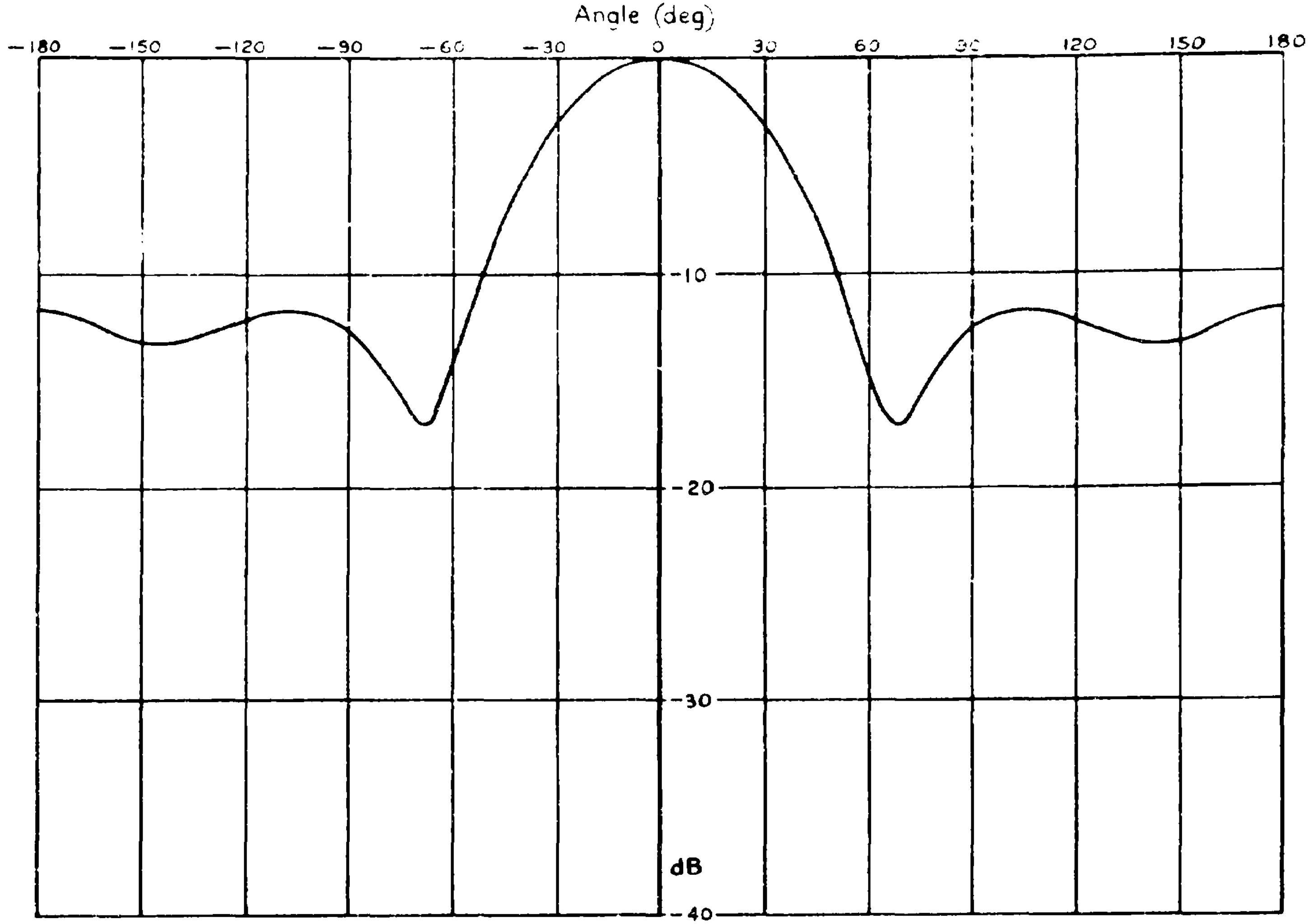

Fig. 10 Optimised pattern for 60° beamwidth

Fig. 11

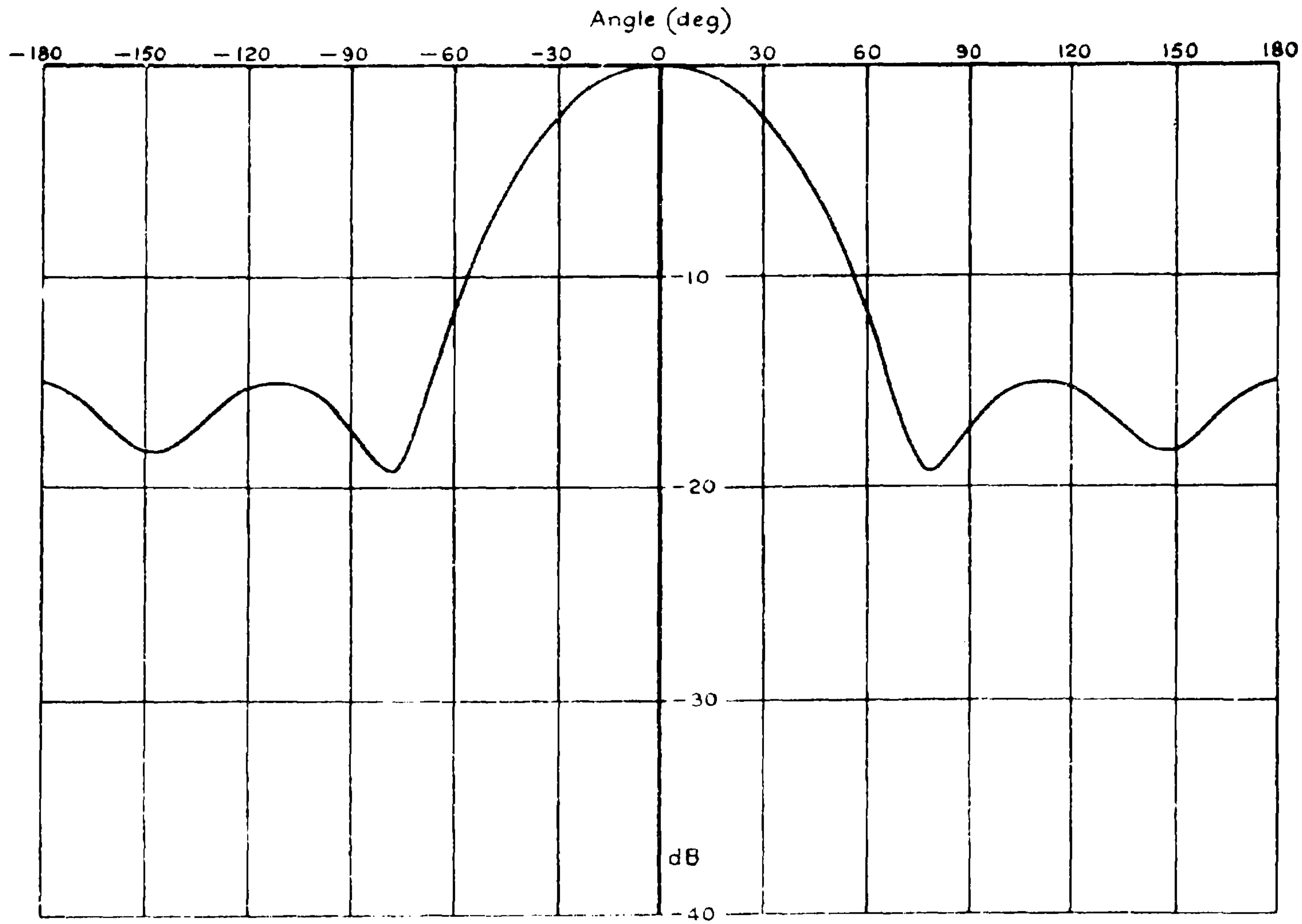

Fig. 11 Optimised pattern for 65° beamwidth

Fig.12

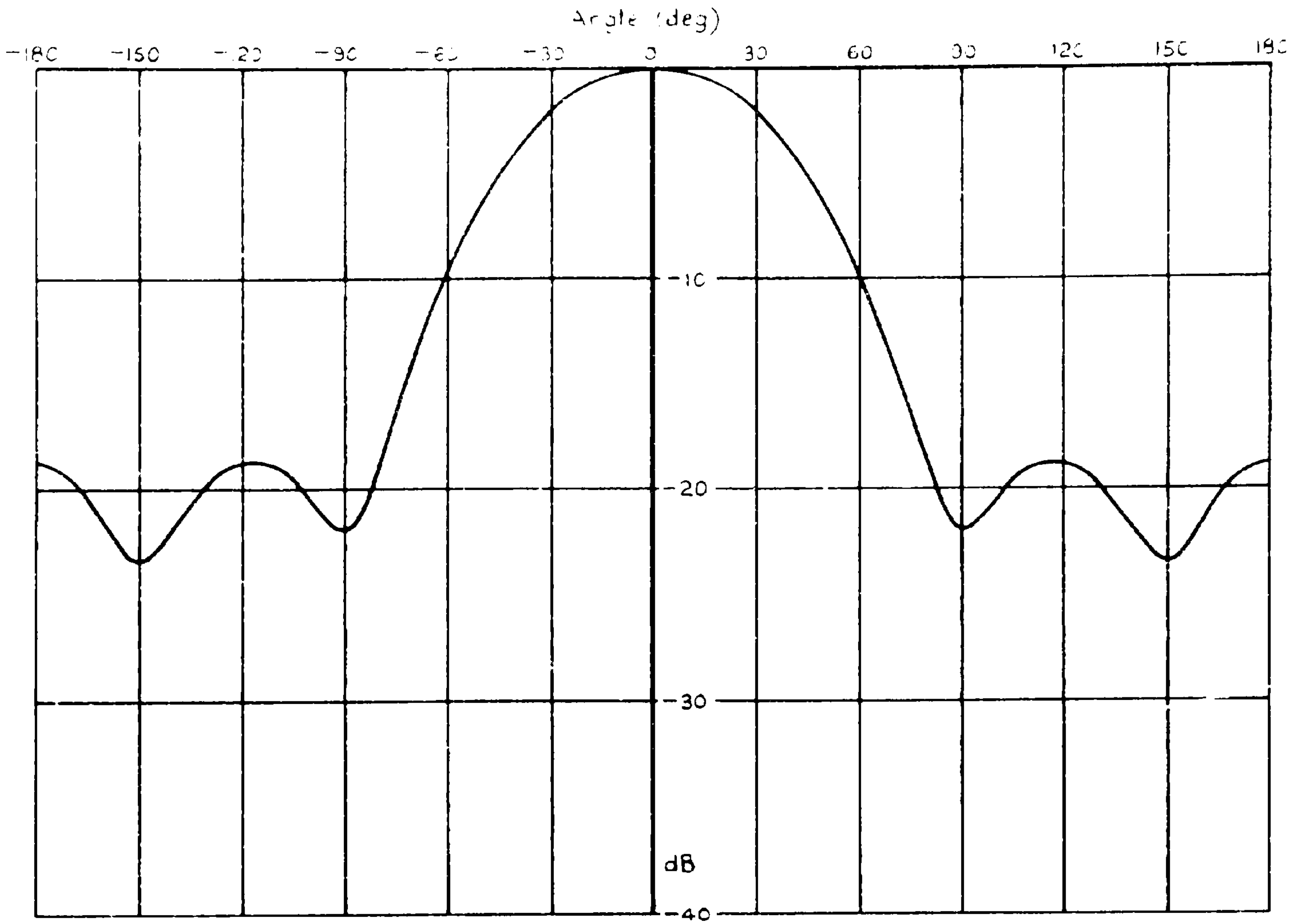

Fig.12 Optimised pattern for 70° beamwidth

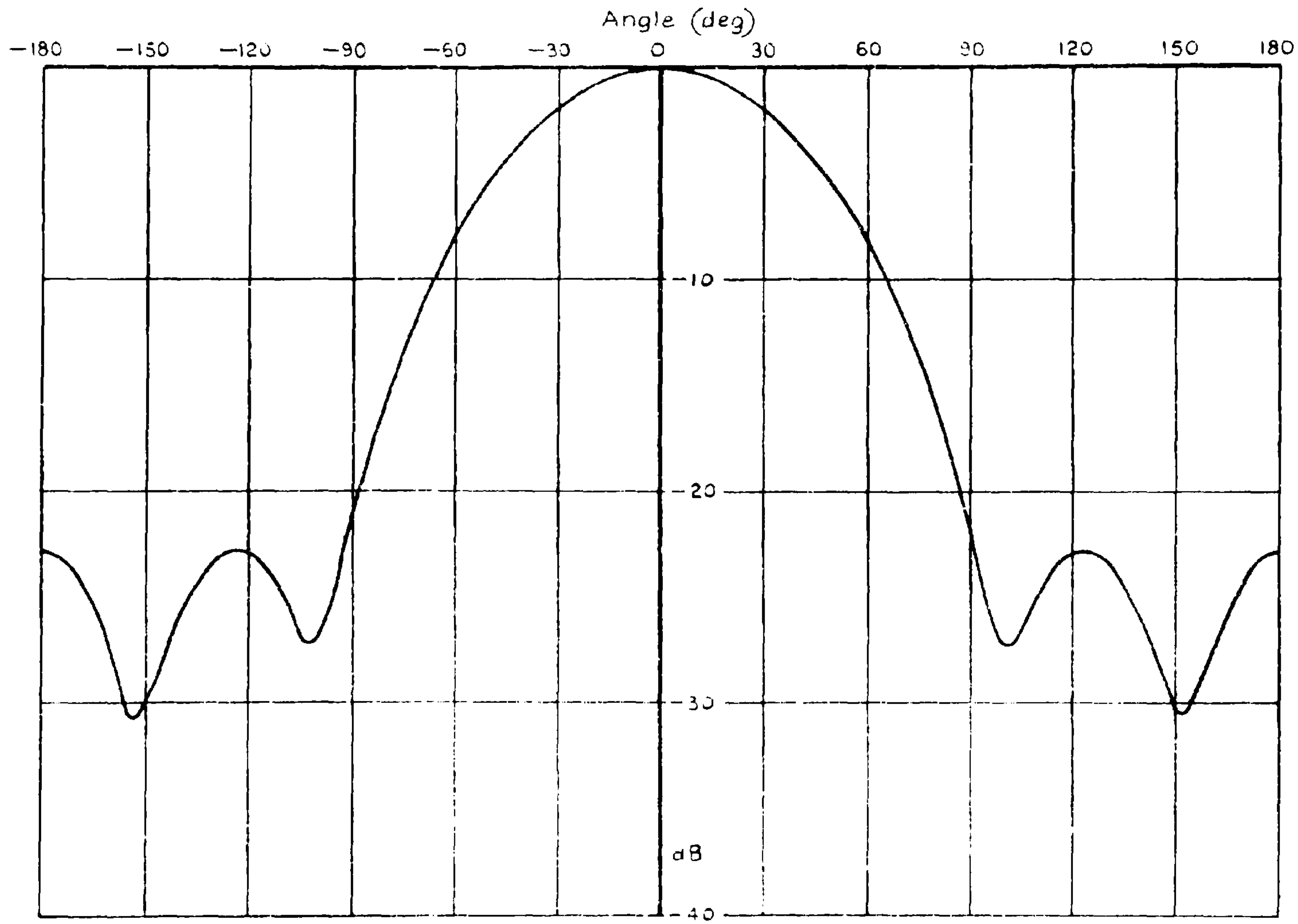

Fig. 13 Optimised pattern for 75° beamwidth

Fig. 13

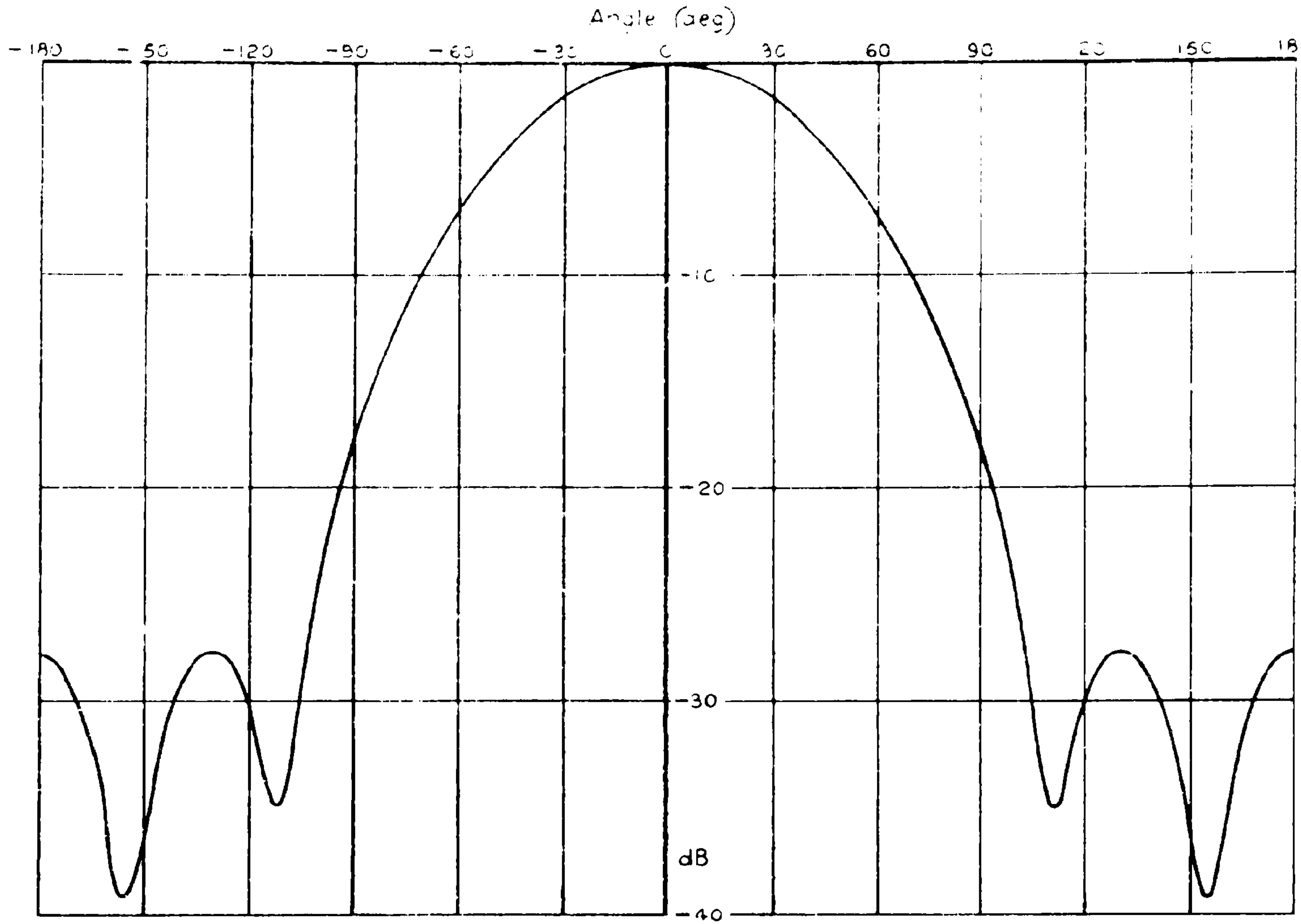

Fig. 14 Optimised pattern for 80° beamwidth

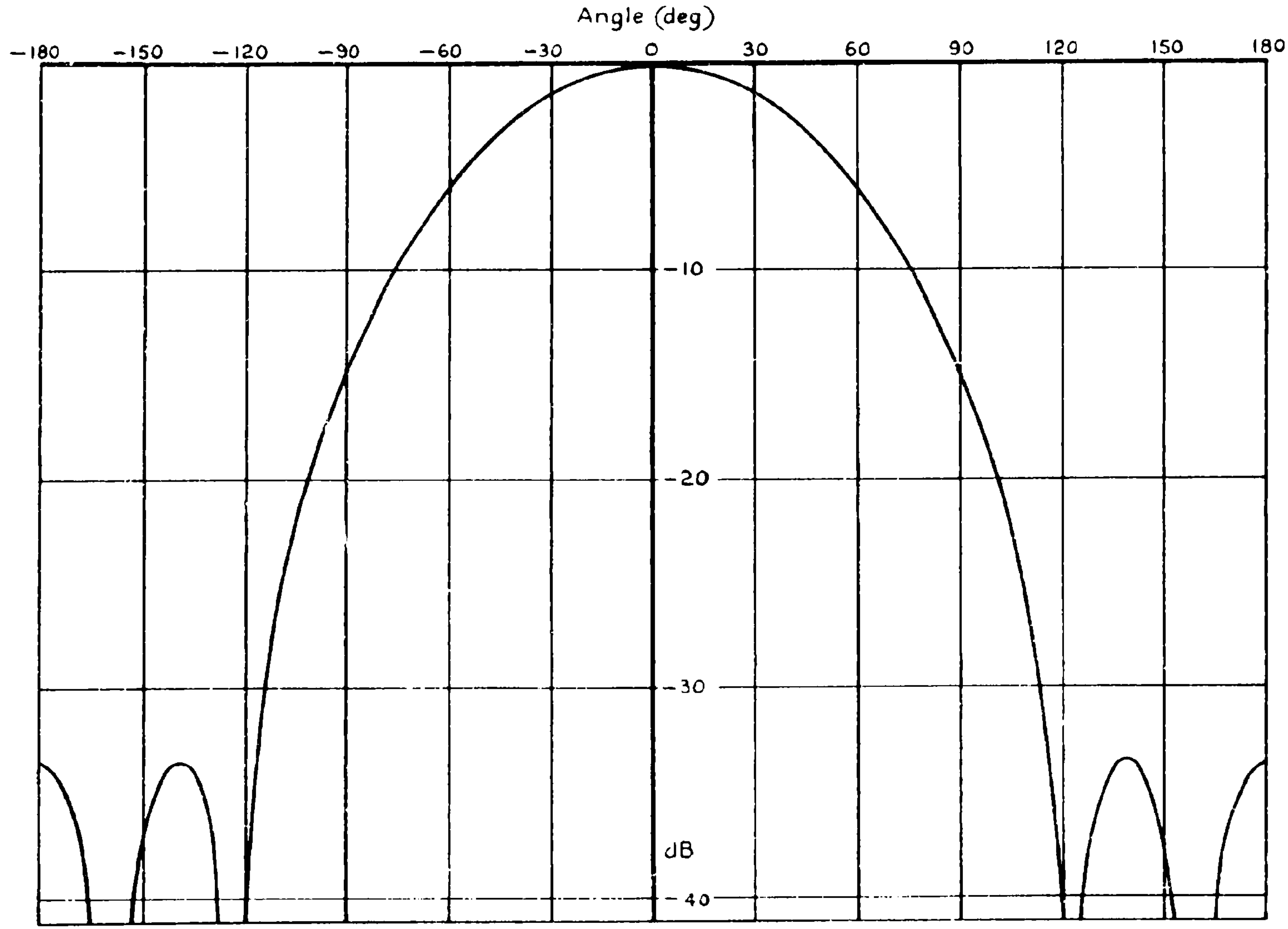

Fig. 15 Optimised pattern for 85° beamwidth

Fig. 15

Fig.16

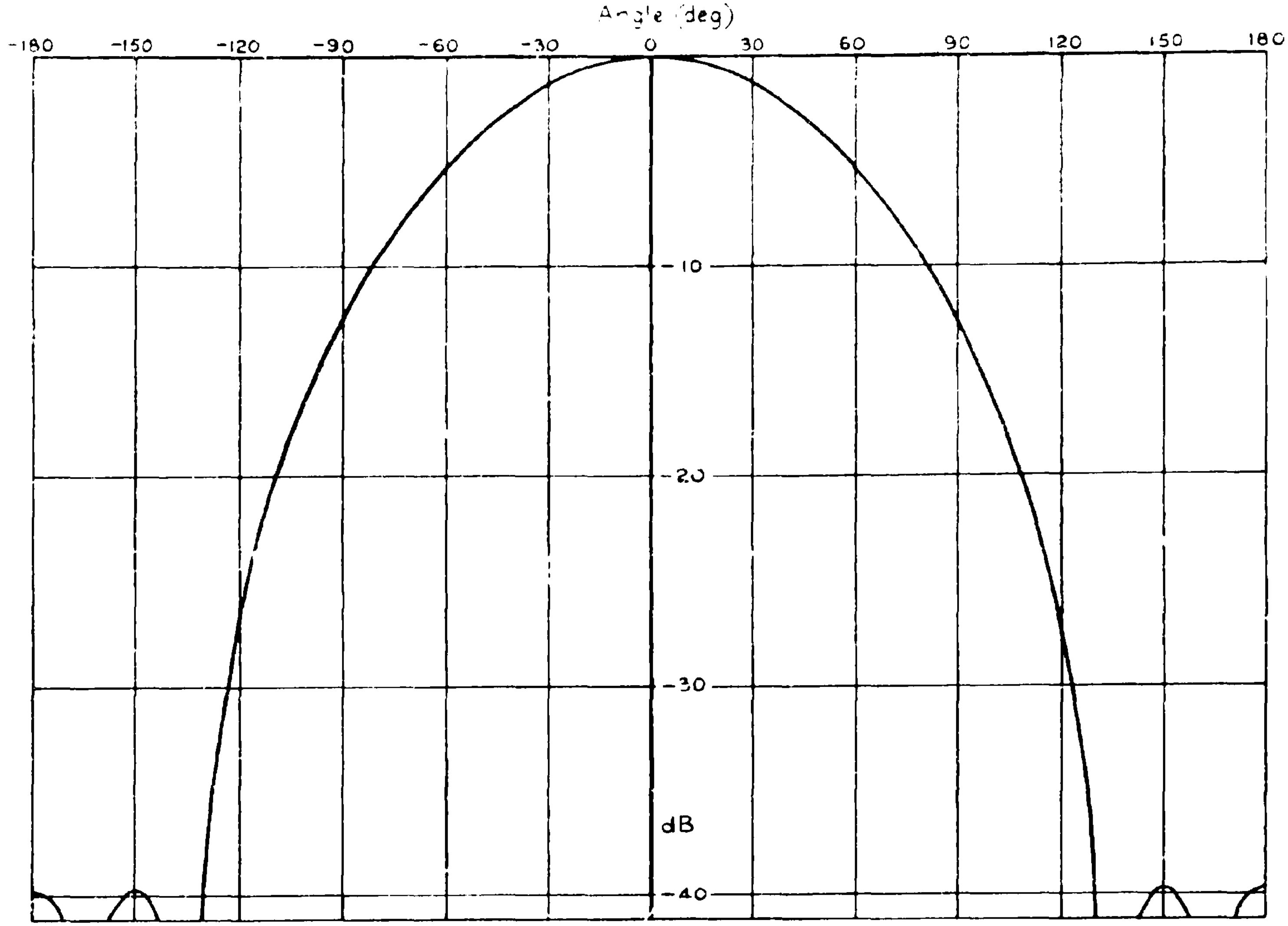

Fig.16 Optimised pattern for 90° beamwidth

008 902491

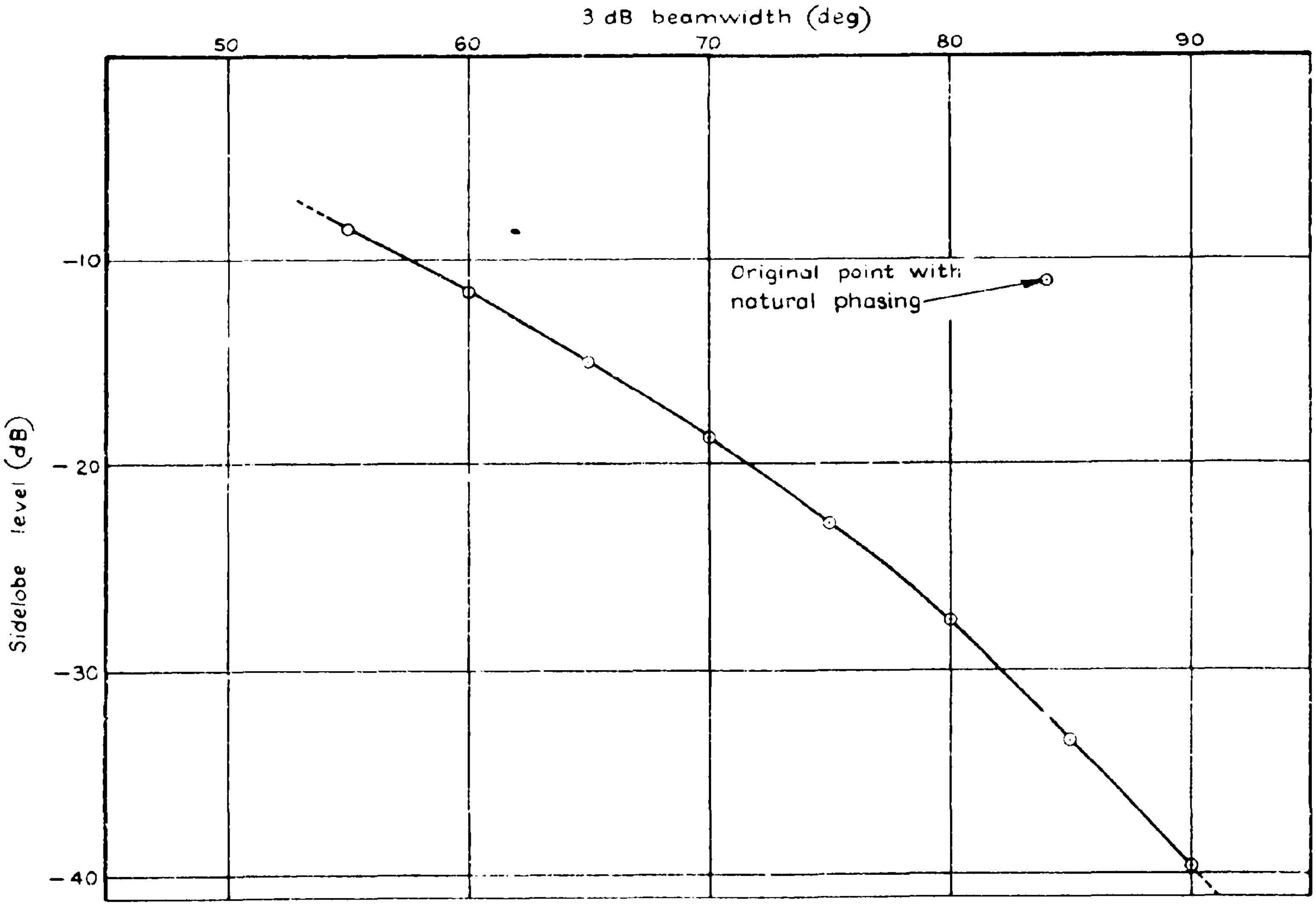

Fig. 17 Relation between sidelobe level and beamwidth

Fig.18

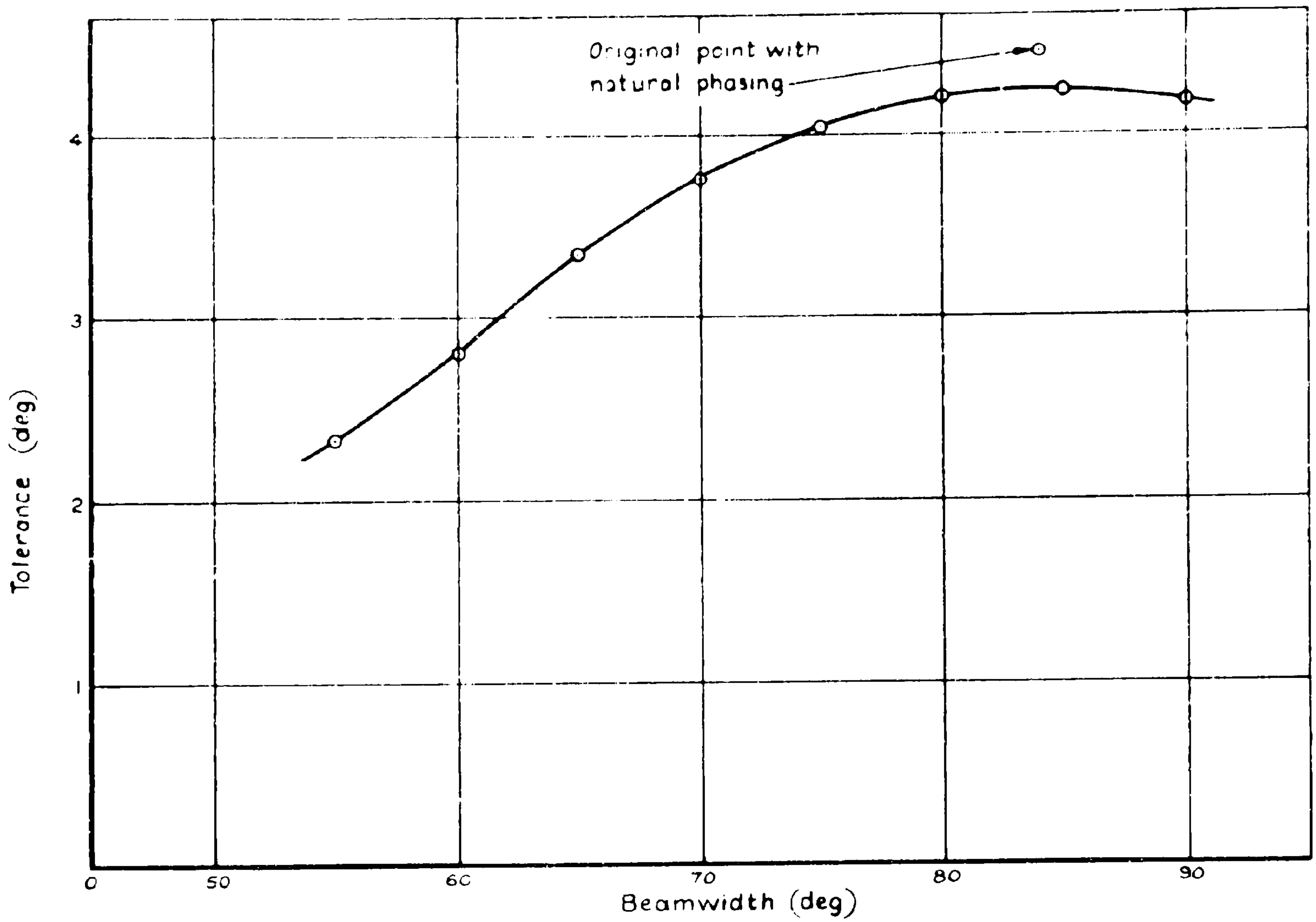

Fig.18 Phase tolerance (−30 dB pattern noise) versus beamwidth